uc3m | Universidad Carlos III de Madrid

Ingeniería Informática

Especialidad en Computación

Curso académico: 2016-2017

Trabajo Fin de Grado

"ARMONIZACIÓN MEDIANTE TÉCNICAS DE COMPUTACIÓN EVOLUTIVA INTERACTIVA"

Carlos Bragado Sánchez

Tutor:

YAGO SAEZ ACHAERANDIO

Presentación: 16/10/2017 – 09:30 horas

Lugar: Escuela Politécnica de Leganés

Aula: 7.2.J02

Agradecimientos.

A María por aguantar incontables horas de mis divagaciones y apoyarme en todo momento.

A mi familia por compartir mi estrés y no dejar que muera de inanición.

A Claudio y Álvaro por su gran apoyo e importante colaboración.

Y, por último, y vital para que la idea se convirtiera en realidad, a Yago por creer en dicha idea y apoyarme en todo lo que ha podido.

BIOGRAFÍA DEL AUTOR.

CARLOS BRAGADO SÁNCHEZ.

Estudiante del grado de Ingeniería Informática con especialidad en Computación de la Universidad Carlos III de Madrid y, actualmente, cursando el Master de Ciencias y Tecnologías de la Informática de esta universidad.

Por otro lado, dedica su vida, de forma partida, entre la ingeniería y sus estudios musicales de guitarra clásica en el Conservatorio Profesional de Música de Leganés.

Ha trabajado como profesor de música, músico profesional y desarrollador web. Recientemente, se publicó el libro “Sistemas Gestores de Bases de Datos e Instalación. UF1469” en cuya escritura ha colaborado.

Su visión de futuro es poder dedicarse a la investigación en el campo de la computación evolutiva.

RESUMEN.

Desde la Grecia antigua se ha perseguido la clave de la inteligencia humana. Los griegos la atribuían al cultivo de la sabiduría en todas sus vertientes (ciencias, letras y artes).
Según han pasado los siglos, esta conexión entre las ramas del saber ha ido desapareciendo. Pero, es en la actualidad, cuando se comienza a abrir un nuevo camino de colaboración.
Parte de este camino se abre con la inteligencia artificial y la búsqueda del concepto de inteligencia, tan preciado por muchos.
Dentro de este campo, reside el de la generación automática del arte, que busca entender y sintetizar la perspectiva humana, frente a estos conceptos, tan comúnmente conocidos como, subjetivos.
Este proyecto busca adentrarse en la música evolutiva, diseñando un algoritmo genético que genere composiciones musicales en el estilo de las Corales de Bach, a partir, únicamente, de las reglas reunidas durante toda la historia de la música sobre este tipo de composición, la síntesis de las acciones de corrección de un maestro de la composición y una línea melódica dada.
Con esto se busca abrir un nuevo horizonte en la generación automática de música, donde prime el ser fiel a la perspectiva del músico e investigar como sintetizarla y adaptarla a los conocimientos computacionales actuales.

ABSTRACT

From ancient Greece the key to human intelligence has been pursued. The Greeks attributed it to the cultivation of wisdom in all its aspects (sciences, letters and arts).
As centuries have passed this connection between the branches of knowledge has disappeared. But, it is now, when it begins to open a new path of collaboration.
Part of this path opens with artificial intelligence and the pursuit of the concept of intelligence, so prized by many.
Within this field lies the automatic generation of art, which seeks to understand and synthesize the human perspective against these concepts so commonly known as subjective.
This project seeks to enter into evolutionary music, designing a genetic algorithm that generates musical compositions in the style of the Bach Corals, starting from only the rules gathered, throughout the history of music, on this type of composition, the synthesis of the corrective actions of a master of the composition and a given melodic line.
This seeks to open a new horizon in the automatic generation of music, where it prevails to be faithful to the perspective of the musician and investigate how to synthesize it and adapt it to the current computational knowledge.

TABLA DE CONTENIDO

01. | INTRODUCCIÓN 9

1.1. | Estructura del documento. 9

1.2. | Planteamiento del problema. 10

1.3. | Algoritmos Genéticos. 12

1.4. | Algoritmos Genéticos Interactivos. 15

1.5. | La música. 15

1.6. | Las Corales de Bach. 17

1.7. | Motivación. 18

1.7. | Objetivos. 19

02. | Estado del arte. 20

2.1. | Una visión general. 20

2.1.1. | Neurogen. 20

2.1.2. | SSEYO Koan 22

2.1.3. | GenJam 22

2.2. | Un enfoque más concreto. 26

2.2.1. | Composición de Sonatas para Piano Mediante Programación Genética Interactiva. 26

2.2.2. | Harmonic Navigator. 28

2.2.3. | Creation of Music Chord Progression Suited for User's Feelings Based on Interactive Genetic Algorithm. 30

2.3. | Extrayendo conclusiones. 31

03. | Análisis del problema. 32

3.1. | Posibles soluciones. 34

3.1.1. | Codificación de las soluciones. 34

3.1.2. | Fitness. 37

3.1.3. | Algoritmo Genético. 40

04. | Diseño de la solución. 43
4.1. | Entorno de pruebas. 43
4.1.1. | Python 3.6. 43
4.1.2. | Eclipse Luna y PyDev. 44
4.1.3. | Microsoft Excel. 45
4.1.4. | Archivo de entrada y salida. 45
4.1.5. | Sibelius, Lilypond y Abjad. 46
4.2. | Algoritmo Genético. 46
4.2.1. | Diseño general. 46
4.2.2. | Codificación de las soluciones. 47
4.2.3. | Operadores de selección. 49
4.2.4. | Operadores de cruce. 50
4.2.5. | Operadores de mutación. 51
4.3. | Estructuras de datos. 51
4.3.1. | Tonality. 52
4.3.2. | Note. 53
4.3.4. | VoicesTessitura. 56
4.3.5. | MusicEncoder. 56
4.3.6. | Individual. 57
4.3.7. | Population. 62
4.4. | Fitness. 65
4.4.1. | Fitness inicial. 65
4.4.2. | Fitness V2: Retoque en las reglas. 69
4.4.3. | Fitness V3: Progresiones Armónicas. 72
4.4.4. | Entrevista con Álvaro Israel Gómez Alvarado. 75
4.4.5. | Fitness Final: Aplicando lo extraído de la entrevista. 78
4.5. | Mejoras sobre el Algoritmo Genético. 82
4.5.1. | Tamaño de población. 82

4.5.2. | Redundancias en la codificación. 83
4.5.3. | Tipo de torneo. 83
4.5.4. | Tipo de cruce simple. 83
4.5.5. | Probabilidad de mutación. 84
4.5.6. | Mejora con Nichos. 84
4.5.7. | Mejora con Islas. 85
05. | pruebas Finales. 86
5.1. | Entrevista con Claudio Tupinambá. 87
06. | Gestión del Proyecto. 91
6.1. | Entorno Socio-Económico. 91
6.1.1. | Educación musical. 91
6.1.2. | Investigación. 92
6.1.3. | Entorno económico. 92
6.2. | Marco Regulador. 92
6.3. | Planificación. 94
6.4. | Presupuesto. 96
6.4.1. | Software. 96
6.4.2. | Hardware. 96
6.4.3. | Viajes y Dietas. 97
6.4.4. | Gastos Indirectos. 97
6.4.5. | Recursos Humanos. 98
6.4.6 | Presupuesto final. 99
07. | Conclusiones. 100
08. | Futuras Líneas de Investigación. 101
09. | Bibliografía. 102
10. | Anexos. 104
10.1. | Índice de figuras. 104
10.2. | Índice de tablas. 105

10.3. | Índice de Gráficas. 106
11. | English's Synopsis. 107
11.1. | Introduction 107
11.1.1. | Document's Structure. 107
11.1.2. | Problem Statement. 108
11.1.3. | Genetic Algorithms. 109
11.1.4. | Interactive Genetic Algorithms. 111
11.1.5. | Music. 112
11.1.6. | The Corals of Bach. 114
11.1.7. | Motivation. 114
11.1.7. | Goals. 115
11.1.8. | Algorithm Design. 116
11.1.9. | Results. 120
11.1.10. | Conclusions. 121

01. | INTRODUCCIÓN

En este documento se encuentra la información referente al trabajo de fin de grado realizado por Carlos Bragado Sánchez bajo la tutoría del profesor/catedrático Yago.

Las páginas siguientes recogen toda la información de dicho proyecto: desde el origen de la idea y el estudio del estado del arte, en el que se encuentra; pasando por el análisis de las soluciones, y el desarrollo de la más viable; culminando con las posibles mejoras y conclusiones, extraídas a lo largo de todo el proceso.

1.1. | Estructura del documento.

Este documento está estructurado con los siguientes apartados:

- **Agradecimientos.**
- **Biografía:** Se explica una breve biografía del autor.
- **Tabla de contenido:** Índice de todos los apartados del documento.
- **Introducción:** Se define el problema propuesto, posteriormente, se explica qué son los algoritmos genéticos, la música y las corales de Bach. Esto permite que el lector se pueda defender en la terminología tratada en el documento.
- **Estado del arte:** Se explica el proceso de análisis de los proyectos e investigaciones dentro del campo de estudio que se está tratando. Se divide en dos partes un análisis general y otro más específico. Con esto, se busca contextualizar el proyecto y relacionarlo con los proyectos analizados.
- **Análisis del problema:** Se explica, de forma detallada, el problema planteado, se exponen posibles soluciones y se argumentan; a su vez, se desgranan y explican detenidamente los objetivos e hipótesis sobre los que se va a investigar.
- **Diseño de la solución:** Se diseña la solución seleccionada entre aquellas analizadas en el análisis, se explica el entorno técnico que del que se va a hacer uso. Se exponen las pruebas realizadas para las decisiones tomadas en el diseño y se presentan las diferentes mejoras desde el primer diseño junto con las pruebas que las refutan.
- **Pruebas finales:** Se exponen y comentan de forma crítica las pruebas realizadas sobre el diseño final de la solución.
- **Gestión del proyecto:** Se detalla la información necesaria dentro del entorno socio-económico, el marco regulador, la planificación llevada a cabo en el proyecto y el presupuesto del mismo.

- **Conclusiones:** Se exponen las conclusiones finales respecto al conjunto global del trabajo realizado y los resultados.
- **Futuras líneas de investigación:** Se plantean posibles extensiones del proyecto: aplicaciones, mejoras, investigaciones derivadas, etc.
- **Bibliografía:** Contiene todas las reseñas de los libros, proyectos y demás documentos utilizados a lo largo del proyecto y, previamente, citados en este documento.
- **Anexos:**
  - **Índice de figuras:** Se resume en un índice todas las figuras del documento.
  - **Índice de tablas:** Se resume en un índice todas las tablas del documento.

1.2. | Planteamiento del problema.

Si uno se remonta a la antigua Grecia, considerada una de las cunas del conocimiento, se observa que, filósofos y grandes sabios de la época, defendían el conocimiento no desde diferentes ramas completamente separadas, sino como la coexistencia y apoyo mutuo entre las mismas.

Se defendía que las personas que dedicaban su vida al conocimiento debían conocer y tener suficiente destreza tanto en la rama de ciencias, como en las de artes y letras.

A lo largo de la historia, se puede comprobar que las principales figuras, bajo las cuales están importantes aportes al conocimiento, eran maestros en las diferentes vertientes de estudio.

Un gran ejemplo de esto es **Leonardo da Vinci**, que realizó aportes a los campos la aerodinámica, anatomía, botánica, pintura, escultura, arquitectura y otros. Entre sus contribuciones más importantes se encuentran “La Gioconda” o los planeadores.

Según ha ido avanzando la ciencia, ha ido creciendo un cisma entre estar corrientes, el más notable es el de la ciencia respecto del arte, cuya colaboración es, prácticamente, inexistente.

Si uno se enfoca en la actualidad se puede observar cómo, cada vez más, se está volviendo a esa concepción de la antigüedad de realizar la búsqueda de conocimiento desde la confluencia de las diferentes ramas; un ejemplo claro es la proliferación de las metodologías ágiles como herramienta de gestión de los diferentes proyectos, donde se trata de conseguir la colaboración de profesionales en distintos campos para llegar a puntos en común de los que poder partir y continuar en paralelo enfocando el proyecto desde diferentes perspectivas pero sobre un mismo camino.

Este concilio entre los diferentes campos es completamente necesario en el ámbito de la inteligencia artificial que, sin entrar en debates a mayores, se puede entender como la búsqueda de llegar a reproducir el concepto de “inteligencia” en las “máquinas”.

El camino hacia este objetivo pasa, necesariamente, por comprender el funcionamiento del aprendizaje de los distintos conocimientos, por comprender el proceso de resolución de los diferentes problemas, etc.

Existe un debate filosófico sobre la definición del concepto de inteligencia; una de las numerosas características, que se le otorga a aquello que llamamos inteligente, es la capacidad de tomar decisiones subjetivas, es decir, el ser inteligente es capaz de sentir y de guiarse por los sentimientos para tomar ciertas decisiones y resolver determinados problemas.

Un ejemplo es el ámbito de la música, en el que se encuentra el problema que se va a tratar en este documento.

Se puede definir la música como una sucesión de sonidos y silencios, que siguen un orden establecido por el compositor y, cuyo fin último, es expresar una emoción en el oyente.

La creencia habitual es que la composición de las obras o canciones es creada a partir de decisiones subjetivas del compositor, aunque también es conocido el uso y necesario aprendizaje de reglas para componer música, pero siempre se tiende a pensar que la música y el arte en general es un producto subjetivo de la creatividad del músico.

Obviamente, es necesario admitir que en cada obra se pueden observar los rasgos de la personalidad de su compositor; pero, consciente o inconscientemente, el artista, busca en su proceso de creación, encajarla en formatos que el ya conoce.

Como David Byrne explica en el primer capítulo de su libro “Cómo Funciona la Música” (Byrne, 2014) [1]:

> *“Los compositores trabajamos al revés, conscientemente o no,*
> *creando obras que encajan en el auditorio del que disponemos”*

Esto también se demuestra en la forma en la que se imparte armonía, análisis y composición en los conservatorios de música.

Existen formas musicales muy estructuradas y regladas utilizadas, principalmente, para la formación de nuevos músicos.

Los ejemplos más importantes provienen del Barroco con la Fuga de escuela, cuyas partes y forma son fijas y fuertemente regladas. Este tipo de composición es una de las utilizadas tanto en las asignaturas de análisis como de composición para iniciar a los alumnos a manejar obras complejas.

Por otro lado, antes de entrar en dichas asignaturas, es necesario que el futuro músico profesional, aprenda Armonía. La forma musical clave para este aprendizaje son las corales de Bach, donde cuatro voces fluyen a lo largo del pentagrama cumpliendo unas normas muy estrictas que permiten al alumno aprender a entender y analizar las diferentes armonías y a controlar los movimientos tanto armónicos como melódicos de las composiciones.

Son las Corales de Bach el centro del problema que se va a plantear e intentar resolver mediante algoritmos genéticos interactivos en las próximas páginas.

Por lo tanto, se va a comenzar con una inmersión en el contexto de los algoritmos genéticos, la música y las corales de Bach.

1.3. | Algoritmos Genéticos.

Para comenzar a hablar de los algoritmos genéticos es necesario conocer su historia y esto lleva al nombre de John Holland, quien establece los cimientos necesarios para que, en los años 70, se desarrollasen dicho método; aunque, hay que remarcar, que no se extiende su uso en problemas reales hasta los años 80.

Los algoritmos genéticos surgen de la idea de imitar las diversas formas de resolver problemas que existen en la naturaleza; más concretamente, se sustentan del teorema de la evolución de Darwin que, de forma muy resumida, explica la evolución de las especies a raíz de la necesidad de adaptación y la supervivencia de los más fuertes y/o mejor adaptados al medio en el que habitan.

De igual forma, estos algoritmos generan una población inicial de individuos, soluciones al problema a tratar, y realizan una selección de los más aptos a dicho problema para que generen descendencia que componga una nueva población de individuos que volverán a pasar por el proceso de selección y reproducción.

De esta forma, el algoritmo va obteniendo cada vez soluciones más adaptadas al problema.

El uso de los algoritmos genéticos se da en la resolución de problemas demasiado extensos y complejos para ser resueltos por otros tipos de métodos ya que obtener una solución por éstos conllevaría un coste computacional y de tiempo inviable.

Esquema general de funcionamiento:

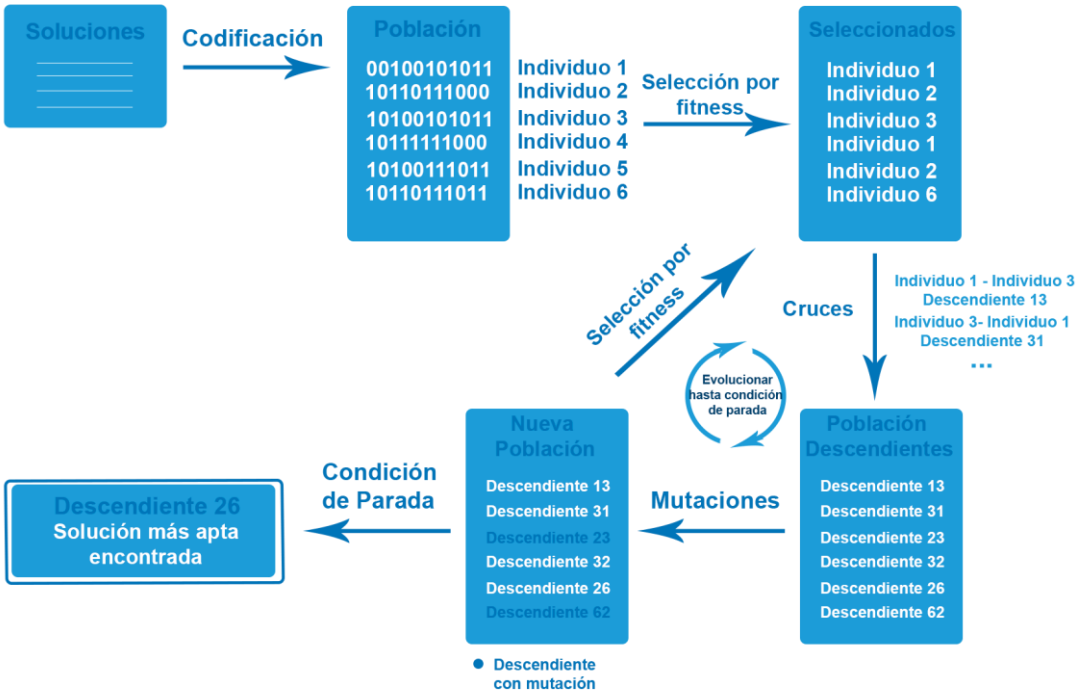


Figura 1 - Esquema general del AG.

Este esquema general de funcionamiento consiste en:

- Primero, se codifican las soluciones formando con cada una un **cromosoma** o **individuo**.
- Estos individuos conforman una **población**.
- Se analiza el grado de adaptación que tiene cada individuo/solución al problema (**fitness**).
- Se realiza una **selección** de los más aptos.
- Sobre esta selección se realizan **cruces** entre ellos con lo que se obtiene una nueva población.
- Se producen ciertas **mutaciones** en los individuos de la nueva población, bajo una limitada probabilidad de que éstas se produzcan.
- Se vuelve a repetir el proceso desde el análisis del fitness de cada individuo de la nueva población hasta que se cumpla la **condición de parada.**
- Finalmente, se devuelve la **solución más apta encontrada** en el proceso de evolución.

Definiciones:

- **Cromosoma o individuo:** Es una posible solución al problema codificada.
- **Gen:** Consiste en cada una de las partes en las que se divide un cromosoma. En el caso de una codificación en bits, un gen es cada uno de los bits de los que se compone el cromosoma.
- **Población:** Conjunto de cromosomas o individuos (posibles soluciones)
- **Población inicial:** Primera población generada en el proceso de evolución.
- **Fitness:** Valor que define el grado de adaptabilidad de un individuo al entorno, es decir, capacidad de una solución de resolver el problema.
- **Selección:** Proceso mediante el que se escoge a los individuos con mejores fitness de la población. Existen diferentes tipos de selección: por torneo, mediante ruleta, etc. Es necesario analizar cuál es más eficaz en cada problema.
- **Cruce o reproducción:** Proceso en el que se forma un nuevo individuo “hijo / descendiente” a partir de la mezcla de genes de dos o más individuos “padres / antecesores” de la población. Existen numerosas formas de reproducción que es necesario analizar para cada problema, puesto que, dependiendo del problema, pueden funcionar unas mejor que otras.
- **Mutación:** Proceso en el que un gen cambia su valor. Esto puede suceder en cada gen de un cromosoma bajo cierta probabilidad, normalmente suele ser una probabilidad muy baja.
- **Condición de parada:** Requisito para que el proceso de evolución termine. Esta condición puede ser desde parar cuando se analice que el algoritmo está estancado en la mejora, hasta detenerse cuando ha pasado cierto tiempo o ciertas generaciones.
- **Generación:** Un ciclo en el proceso de evolución, es decir, si se observa el esquema mostrado anteriormente, cada vez que se llega a “Nueva Población” se puede nombrar como el inicio de una nueva generación.

  Innformación extraida de los apuntes de la asignatura Algoritmos Genéticos Evolutivos de la Universidad Carlos III de Madrid y del libreo “Aprendizaje Automático” [2]

1.4. | Algoritmos Genéticos Interactivos.

Estos algoritmos funcionan de la misma forma que los ya explicados, la gran diferencia es la necesidad de la colaboración humana en el proceso evolutivo.

La interacción más frecuente se encuentra en el fitness, donde es el humano quien evalúa cada individuo (por ejemplo, en fitness entendidos como subjetivos, ya sean de obras de artes, literatura, música, etc). También, como se verán en el apartado de Estado del arte, se puede producir la interacción en la creación de una base de datos de la que obtener una población inicial. [3]

1.5. | La música.

Se puede definir la música como una sucesión de sonidos y silencios que siguen una estructura armónica, melódica y rítmica. Con la composición de una obra se busca expresar algún tipo de sentimiento al oyente.

Como se ha explicado, la música se compone de:

- **Pentagrama:** Conjunto de 5 líneas y 4 espacios donde se escriben las diversas figuras musicales.
- **Nota:** Elemento que sirve para dar nombre a cada uno de los sonidos utilizados en la música. Son 7: Do – Re – Mi – Fa – Sol – La – Si
- **Semitono:** Distancia mínima entre dos sonidos que se distinguen a diferente altura. Es necesario añadir esto último ya que culturas como la árabe se utilizan los cuartos de tono.
- **Tono:** Distancia entre dos sonidos equivalente a dos semitonos.
- **Alteración:** Símbolo situado a la izquierda de una nota (en la escritura en pentagrama) y que indica que dicha nota tendrá un sonido un semitono por encima (sostenido “#”) o por debajo (bemol “b”) del habitual.
- **Becuadro:** Alteración que representa que el sonido de la nota que acompaña es el habitual / natural. Se utiliza a la hora de quitarle una de las otras alteraciones a la nota o de recordar que ha de sonar natural.
- **Clave:** Figura situada, normalmente, al inicio de cada pentagrama que indica la disposición de las notas. Por ejemplo, en clave de sol en segunda línea indica que la nota sol está situada en la segunda línea. Existen cuatro nomenclaturas: Clave de Sol, Clave de Fa y Clave de Do. A su vez, la clave de sol siempre es en segunda línea, la clave de Fa puede ser en 4ª o 3ª línea y la clave de Do puede ser en 1ª, 3ª o 4ª línea.

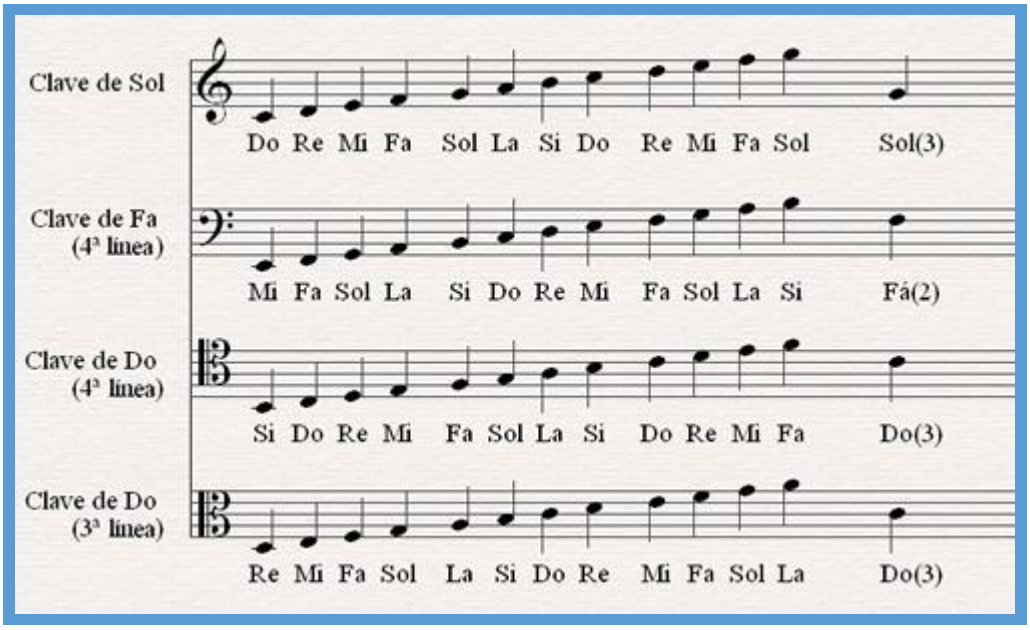


Figura 2 - Claves musicales

- **Octava:** Indica el nivel de grave o agudo en el que se encuentra una nota. En este proyecto se van a contemplar cuatro octavas. Como se puede ver en figura anterior las últimas notas van acompañadas de un número que indica en la octava en al que se encuentran. Esto también ejemplifica la utilidad de las claves, al comprender entre sus notas, que figuran dentro del pentagrama, diferentes octavas entre las diversas claves.

- **Intervalo:** Es la distancia que existe entre dos notas. Se compone de el número de notas que hay entre ellas y, dependiendo del número de tonos y semitonos que se encuentren entre ambas, se nombraran como **Mayores** o Justos, en el caso de los intervalos de 4ª, 5ª y 8ª; estos cumplen la distancia tomada desde do natural a la nota en el intervalo correspondiente. **Menores**, el intervalo contiene un semitono menos que la referencia desde do natural. **Aumentado,** el intervalo contiene un semitono más. **Disminuido**, el intervalo contiene un tono menos.

- **Armonía:** Es el estudio de la estructura de las notas que se emiten al mismo tiempo.

- **Melodía:** Sucesión de notas emitidas una detrás de otra.

- **Acorde:** Composición de 3 o más notas emitidas al mismo tiempo.

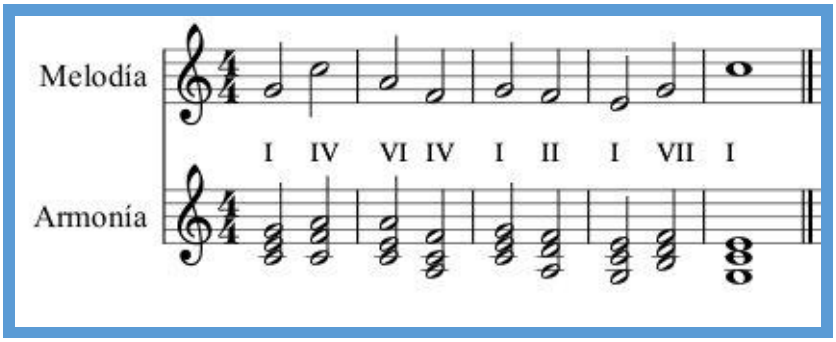


Figura 3 - Acordes

- **Cifrado:** Símbolos utilizados para designar los tipos de acordes. En este proyecto se va a utilizar el cifrado compuesto por cifras árabes, que es el estándar dentro de la música clásica. [4]
- **Cadencia:** Conjunto de dos o tres acordes que dan cierta sensación de finalización. Ejemplo: Cadencia Perfecta entre los acordes V y I.
- **Escala:** Sucesión de siete notas siguiendo el orden mencionado anteriormente, pero si la escala es de Re se comenzará por dicha hasta volver a llegar al Re.
- **Tonalidad:** Indica las alteraciones que debe seguir una escala.
- **Armadura:** Generalmente situada al principio del pentagrama, compuesta por sostenidos o vemos, e indica el tonalidad en la que se encuentra la obra.

1.6. | Las Corales de Bach.

En su esencia es una coral protestante, forma musical compuesta por cuatro voces. Las primeras obras se encuentran remontándose al siglo XVI en la iglesia Luterana y eran utilizadas en ceremonias religiosas.

Johann Sebastian Bach (1685 – 1750) es el compositor de corales más conocido y por ende reciben su nombre. [5]

Debido a las rigurosas reglas de composición que se debe seguir para crear una obra en dicha forma musical y, a su vez, la simpleza de la rítmica en la que apenas se aprecia variación, se ha adquirido como herramienta de armonía en los conservatorios profesionales de música.

Las reglas de composición que se tendrán en cuenta en este proyecto, hacen referencia al primer curso de armonía de los conservatorios españoles.

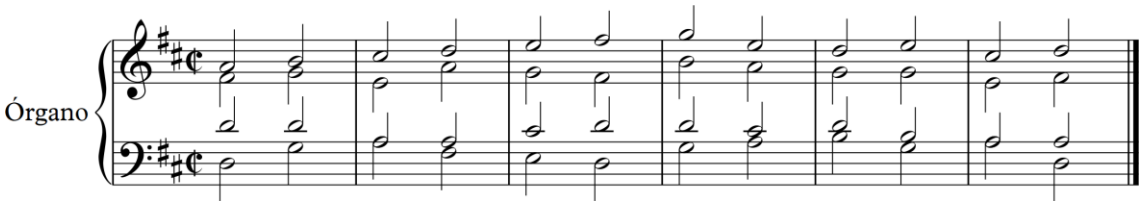


Figura 4 - Coral de Bach

Como se puede ver en la imagen, los Corales de Bach están compuestos de cuatro voces: la más grave es el **bajo**, por encima está el **tenor**, seguidamente se encuentra el **contralto**, y la voz más aguda recibe el nombre de **Soprano** o **Tiple.**

1.7. | Motivación.

El origen de este proyecto viene motivado desde la elevada formación, en el conservatorio profesional, de música del autor.

En una iniciativa propia de buscar el enlace de los estudios de ingeniería informática con los musicales, surge la idea de buscar ese punto de unión por medio de los algoritmos genéticos.

La idea de algoritmo genético es conseguir que genes, que por sí solos, no pueden resolver nada excesivamente complejo, se agrupen en cromosomas, que codifiquen posibles soluciones a problemas complejos, y éstos, a su vez, convivan con otros cromosomas en poblaciones buscando crear nuevas generaciones más adaptadas a la solución.

El autor aquí observa un similitud con la música, los genes se pueden asemejar a las notas de una partituras, por si solas no expresan nada concreto, sin embargo al juntarlas en melodías, que serían los cromosomas en los algoritmo genéticos, comienzan a dar sentido a la obra. Estas melodías se agrupan formando armonías que dotan de una expresión y coherencia completa a la composición.

Una vez vista este punto de enlace, el deseo era que esta investigación pudiera ser útil no sólo de cara al campo de la computación genética, sino en la enseñanza de la música.

Es por esto que se plantea poder llevar esta técnica a alguno de los problemas planteados en las diversas asignaturas de composición.

Viéndolo como la primera parte de un proyecto de investigación a gran escala, se ha creído conveniente empezar por la asignatura de armonía y las Corales de Bach; que, como ya se han explicado, conllevan una normativa de composición muy rigurosa y a su vez su composición es sencilla y fuertemente estructurada.

Por último, remarcar la gran utilidad que puede tener toda la implementación de la normativa musical de las Corales de Bach de cara a otros proyectos musicales y como gran apoyo en la enseñanza de dicha forma musical, al proporcionar soluciones y una posible herramienta de corrección.

1.7. | Objetivos.

El objetivo principal que se propone resolver este proyecto, es llegar a diseñar un algoritmo genético capaz de encontrar soluciones viables a ejercicios de Corales de Bach.

Concretando más en profundidad, el algoritmo debe recibir una línea melódica que haga referencia a la primera voz de la coral o soprano, o a la cuarta voz o bajo. Generalmente, los primeros ejercicios que abordan los alumnos son con la voz del bajo dada, pero en el final del aprendizaje la entrada corresponde a la voz soprano.

Una vez recibida la línea melódica, el algoritmo deberá buscar soluciones que la armonicen en cuatro voces contando la dada (bajo, tenor, contralto, soprano). Estas soluciones deberán ser evaluadas por un fitness que contemple la normativa de las Corales de Bach y los puntos positivos que tienen aquellas que cumplan todas las normas.

Finalmente, como fin de este proceso, se analizarán de forma crítica los resultados buscando conclusiones a cerca del algoritmo diseñado.

02. | ESTADO DEL ARTE.

En las últimas décadas se han desarrollado un gran número de proyectos e investigaciones referentes a la composición musical generada automáticamente. Una gran parte de estos trabajos se han enfocado en las redes de neuronas o en los modelos de Márkov, como es el caso de EMI (Experiments in Music Intelligence) [6] que, a partir una base de datos de trabajos musicales, entrena un modelo de Márkov; de esta forma se consigue componer obras de forma automática en el estilo de la música que contenga la base de datos de entrenamiento.

Por otro lado, estos proyectos se han enfocado hacia la composición evolutiva, es decir, aquellos compositores automáticos que en sus entrañas se nutren de algoritmos genéticos. Se puede comprobar en libros como “Evolutionary Computer Music” [3] que existen numerosas formas de afrontar dicho tipo de técnica de composición automática. Desde la improvisación de melodías o acompañamientos, como se verá en el proyecto **GenJam** [7] hasta algoritmos que permiten crear obras imitando el estilo de compositores u otras referencias.

Para esta investigación, es necesario centrar el análisis del estado del arte en aquellos proyectos que usen dichas técnicas de computación evolutiva en la composición de obras musicales.

2.1. | Una visión general.

En una primera fase, se ha analizado, de forma general, proyectos con mucha repercusión en este campo. Concretamente en 3 que se cree que dejan ver las perspectivas más importantes del campo de estudio:

- **Neurogen:** Cuyo análisis aporta una visión respecto al uso de algoritmos híbridos (algoritmos genéticos y redes de neuronas).
- **SSEYO Koan:** Proporciona la perspectiva comercial de este tipo de trabajos, demostrando que todas las investigaciones realizadas en el campo son incorporables y aceptadas en la comunidad de producción musical.
- **GenJam:** Un proyecto inspirador para el autor de esta investigación, de gran valía como aporte al campo de la música evolutiva y del que se extraen muchas ideas para la investigación plasmada en este documento.

2.1.1. | Neurogen.

Realizado en 1991 por P.M. Gibson y J.A. Byrne, considerado uno de los primeros desarrollos de la composición musical automática. Mezcla el uso de redes de neuronas junto con algoritmos genéticos y con la interactividad humana.

Las redes de neuronas las utiliza para generar una heurística a partir del aprendizaje de numerosas obras de música occidental tradicional calificadas como malas o buenas por experimentación con personas reales.

Esta heurística es utilizada como fitness para evaluar los diferentes fragmentos que genera el algoritmo genético.

Este algoritmo híbrido se limitó a pequeñas composiciones en Do Mayor estructuradas en cuatro partes correspondientes cada una a una melodía que conforman la armonía de la obra, pero sus autores concluyen dejándolo abierto a la posible mejora con algoritmos genéticos múltiples y una reconstrucción de la red neuronal utilizada. [8]

En cuanto al diseño del algoritmo genético, los individuos son elaborados por cadenas de bits en las que 1 significa la existencia de un sonido y 0 de un silencio. Una vez realizada la codificación de los individuos la red de neuronas, previamente entrenada, como se ha comentado, los califica en función de la heurística que ha generado en su aprendizaje. Posteriormente, el algoritmo genético se sirve de un modelo de selección por el algoritmo de Monte Carlo, los cruces se realizan a nivel de bit aleatorio y luego se realizan las mutaciones que surjan. Es necesario remarcar que primero se realiza el proceso para obtener los ritmos y luego para las melodías, terminando con un generador de armonías para unir todo. [9]

Un esquema simple de este proceso sería:

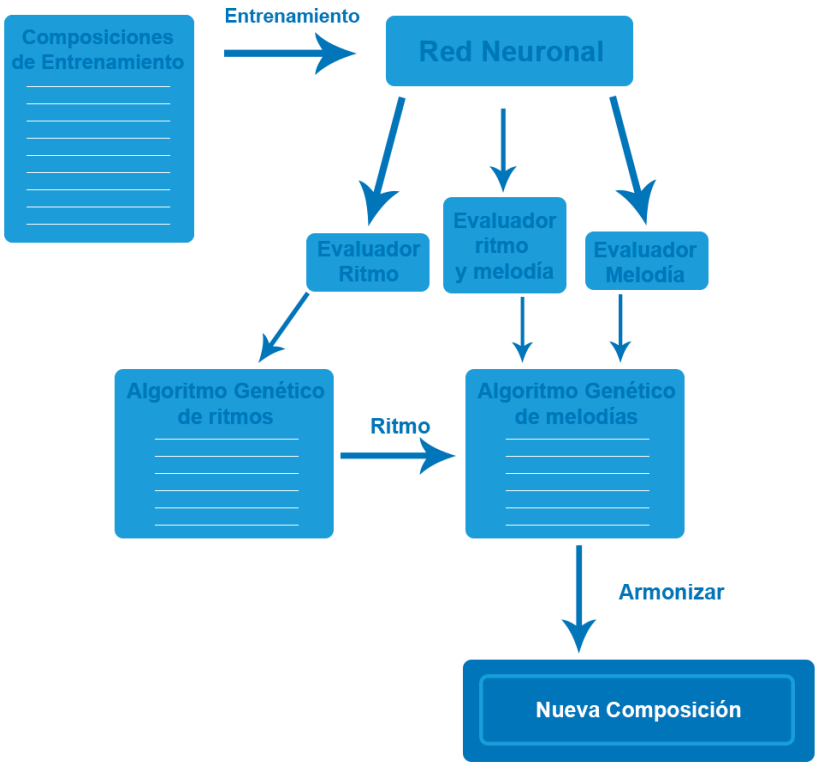


Figura 5 - Neurogen esquema

En este caso, como pasa a lo largo de este tiempo, es necesario apoyar el algoritmo genético con una red de neuronas, y ésta, es demasiado poco robusta para generar este tipo de composiciones por sí misma. Por otro lado, al usar la red de neuronas como fitness limita en gran medida el control de la composición, lo que da lugar a las grandes limitaciones que se explican en el proyecto. Por ello, en esta investigación se ha propuesto usar única y exclusivamente algoritmos genéticos para crear obras musicales y se busca el obtener un fitness enfocado a evaluar de forma matemática todos los fallos armónicos y melódicos que pueda tener la composición, generando a la vez melodías y armonías (lo que en música se conoce como crear el contrapunto).

2.1.2. | SSEYO Koan

Koan es considerado uno de los primeros sistemas de generación musical en tiempo real. Se empieza a desarrollar en 1990 por Pete y Tim Cole junto con Brian Eno quien lo presenta en 1994.

Es en 1995 cuando Brian Eno lo utiliza para sacar el disco “Generativemusic 1”.

En 2007 pasa a estar dentro de la herramienta “Intermorphic Noaltikl” [10]

Y, finalmente, en 2017 se incluye dentro del software “Wotja” [10]

Esta herramienta supone una revolución en el mundo de la creación musical ya que permite controlar unos 50 parámetros para crear la obra.

No se ha podido encontrar mucha más información acerca de su funcionamiento, puesto que es una herramienta muy potente que se comercializa y la empresa mantiene dicha información privada, pero se ha creído conveniente darle una mención como símbolo de la gran utilidad que puede ofrecer las investigaciones en este campo.

2.1.3. | GenJam

John A. Biles [7] es músico de jazz e investigador dentro del campo de la unión de las nuevas tecnologías a la música. Ha generado numerosas contribuciones a dicha área de estudio. Se comenzará explicando una de las más conocidas “GenJam” contenida en su libro “Evolutionary Computer Music” [3]

GenJam (Genetic Algorithm Jammer y Genetic Algorithm Jazz Jam Session) [7] se basa en un algoritmo genético para componer solos y acompañamientos de jazz en tiempo real. Su primera versión fue presentada en 1994.

El mismo define este algoritmo, en su charla TEDx [11], como un alumno de improvisación de jazz; de forma que a raíz de generar piezas de este estilo, un maestro le devuelve una evaluación a cerca de aquello que ha generado.

En principio, el concepto es el obtenido por numerosos proyectos de esta índole; la genialidad reside en el camino que sigue el diseño de su algoritmo genético para conseguir su objetivo.

Se puede ver resumido en un pequeño esquema extraído de “Evolutionary Computer Music” [3]

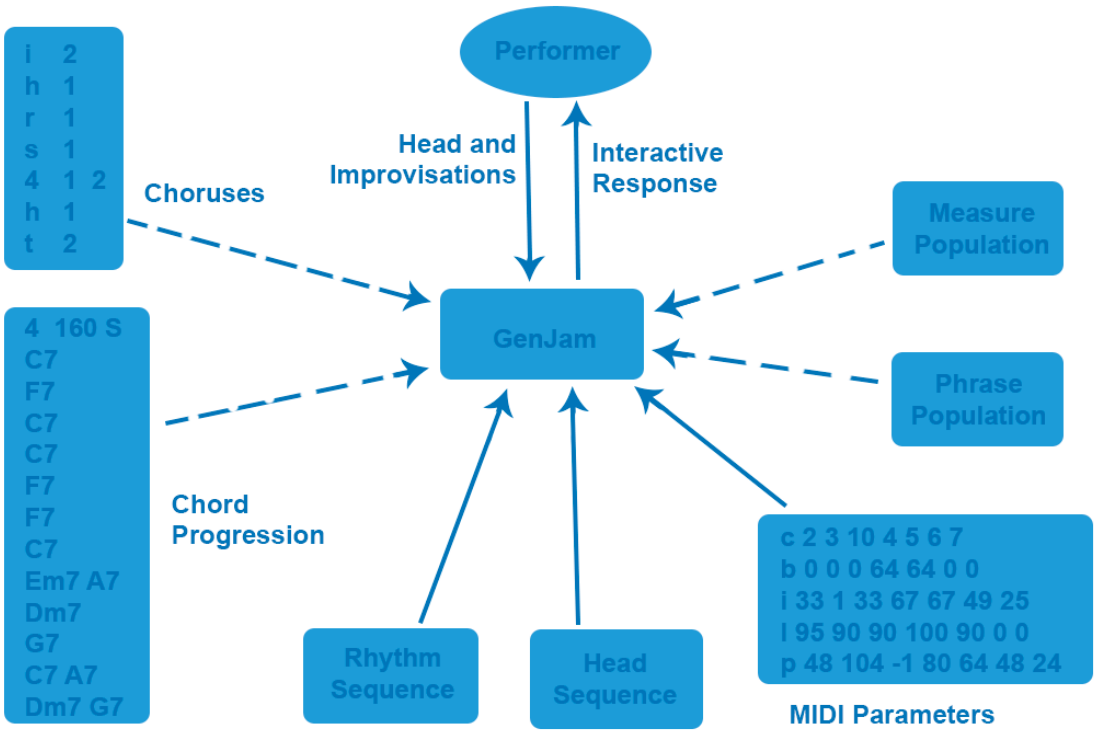


Figura 6 - GenJam

Como se puede observar el algoritmo trabaja con dos poblaciones, una para las medidas de tiempo (**Measure Population**) y otra para las melodías (**Phrase Population).** Algo parecido a lo explicado con Neurogen pero, en vez de usar diferentes algoritmos genéticos para los ritmos y las melodías, usa poblaciones diferentes en un mismo algoritmo.

La forma de relacionar estas dos poblaciones es que un cromosoma de la población de melodías está ligado con un cromosoma de la población de tiempo y a su vez ésta es definida por los parámetros MIDI introducidos.

Estos parámetros MIDI (**MIDI Parameters**) dan indicaciones al algoritmo sobre 35 direfentes variables para cada uno de los 7 canales MIDI disponibles (número del canal, volumen, etc)

Cabe destacar, que el algoritmo usa las poblaciones completas como solución a evaluar de forma interactiva.

Por otro lado, se le introduce una progresión de acordes (**Chord Progression**) que también contiene la información sobre en qué octava moverse, el tiempo, y la medida de las 8 notas.

Los coros (**Choruses**) le indican qué debe hacer para cada una de las repeticiones de la forma definida en la progresión de acordes.

GenJam también precisa de un “background” de información sobre la rítmica en la que debe inspirarse, para esto, se le introduce un archivo MIDI con 5 canales (bajo, batería, cuerdas, guitarra y piano) por medio de la secuencia rítmica (**Rythm Sequence**).

Y, finalmente, recibe un segundo archivo MIDI con melodías, líneas de bajo, ritmos de batería, etc. Resumiendo, este archivo introducido en “**Head Sequence**” le proporciona al algoritmo un “background” en el campo armónico.

Si se centra el interés en el diseño de la **codificación de los individuos** en las dos poblaciones, se observa que la **población de melodías** tiene un tamaño de 48 individuos; cada uno contiene su fitness correspondiente y 4 índices que lo vinculan con 4 individuos de la **población de tiempo**; dicha población tiene dimensión de 64 individuos en cada uno se indica el fitness y las notas que contiene. Estas notas están codificadas del 1 al 14, limitando el rango a dos octavas y a su vez las posibles notas están limitas a las propias de él acorde de la progresión en el que se encuentre, el 0 indica silencio y el 15 indica que se alarga la medida de la nota anterior. Para entender esto último se explica que cada nota está bajo la medida de corchea de forma predefinida, entonces si el individuo contiene un 1 indica que se debe tocar un Do corchea, sin embargo si después se contiene un 15 ese Do se alarga a una negra que es equivalente a dos corcheas.

Por último, indicar que, como se ha comentado anteriormente, cada individuo de la población de melodías se compone de cuatro individuos que la población de tiempo, que se corresponden, cada uno con un compás de la composición. La suma de los 11 individuos de la población de melodías, proporcionan la composición completa que es evaluada de forma interactiva.

Un ejemplo de todo esto, extraído del libro “Evolutionary Computer Music” [3]

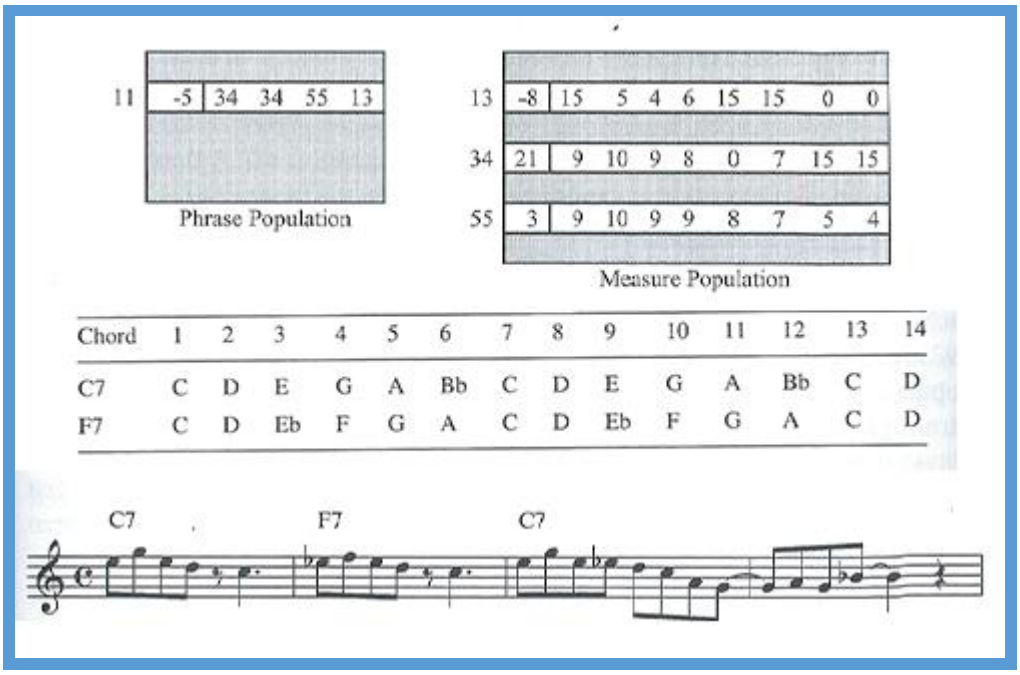


Figura 7 - Codificación GenJam

Respecto a los operadores genéticos, las **mutaciones** las genera de tres tipos:

- **Mutaciones en la población de tiempos:** Estas mutaciones consisten en invertir el orden de los genes, rotaciones, transponerlos, etc.
- **Mutaciones en la población de melodías:** Estas mutaciones consisten en tareas de diferentes tipos de reorganización o cambios en los genes e incluso en sustitución de la melodía por una nueva.

El cruzamiento utilizado es el habitual cruce en un único punto elegido de forma aleatoria.

Una vez visto, de forma general, el funcionamiento de este algoritmo, se puede extraer del mismo las limitaciones de su codificación a la hora de permitir únicamente dos octavas de notas; que, como se verá en la solución del problema que compete a esta investigación, se ha intentado solventar cambiando a otro tipo de codificación.

Por otro lado, la idea de usar poblaciones como una población en sí misma y a su vez usar dos poblaciones diferentes enlazadas, es una propuesta muy interesante, pero se cree que es poco conveniente en el caso de las Corales de Bach ya que no genera melodías sobre un único acorde sino que la propia progresión de acordes forma la melodía y, dicha progresión, ha de ser creada, no se da *a priori*.

Si se va a utilizar el cruzamiento mencionado en GenJam, con una pequeña variación, ya que se utiliza en dos puntos en vez de uno y las mutaciones se entienden excesivas para el formato de las corales de Bach, ya que, en GenJam, son muy útiles debido al objetivo de improvisación y al carácter compositivo del estilo del Jazz.

2.2. | Un enfoque más concreto.

Una vez se ha analizado, de forma general, ciertos rasgos de la generación automática de música mediante el estudio de grandes aportaciones al campo de estudio; se van a desgranar otros proyectos más enfocados al problema en estudio, para hacer una comparativa con los objetivos propuestos en este proyecto y las posteriores soluciones y remarcar las ideas extraídas de los mismos, así como las descartadas.

2.2.1. | Composición de Sonatas para Piano Mediante Programación Genética Interactiva.

Este proyecto lo han desarrollado Flor Del Carmen Herrera Juárez, Karina Peña y Perfecto Malaquias Quintero Flores, integrantes del laboratorio de investigación en Tecnologías Inteligentes del Departamento de Sistemas y computación del Instituto Tecnológico de Apizaco, México.

El proyecto consiste en aplicar algoritmos genéticos interactivos para crear obras musicales para piano en la forma Sonata.

En primer lugar, se sirven de un descriptor gramatical para generar un árbol que codifique las sonatas y asegurar que mantienen su estructura de forma correcta según la teoría de la composición clásica de la forma Sonata.

Un individuo se corresponde con una sonata bajo la codificación en árbol:

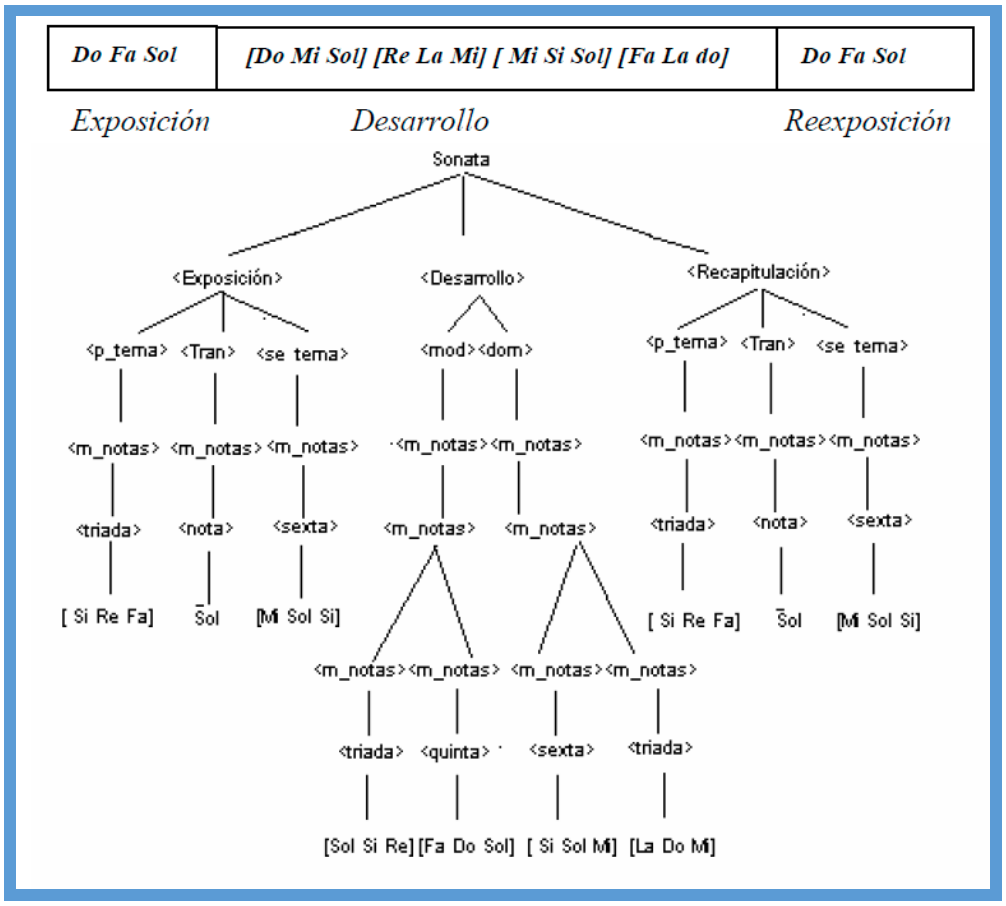


Figura 8 - Sonata en árbol.

Los cruces los realizan en 3 partes, se cruza la exposición, el desarrollo y la rexposición. El cruce de la exposición y la re exposición debe ser el mismo ya que ambas partes deben ser iguales en la forma Sonata.

El fitness de cada individuo es obtenido mediante un autómata, que se encarga de comprobar la corrección de la gramática bajo la que se codifica el individuo. Por otro lado, se le añade la opción de que las sonatas sean valuadas por un sujeto humano, aplicándole, de este modo, la interactividad al algoritmo.

Usando poblaciones de 12 individuos con tamaños diferentes los resultados que obtienen bastante buenos y eficientes ya que las pruebas duran entre dos y cinco minutos cada una hasta obtener resultados válidos. [12]

Se ha decidido mostrar este proyecto debido al interés que puede suscitar el uso de codificación en forma de árboles. Pero también muestra una amplia limitación debido a que la gramática restringe en gran medida las combinaciones y no da pie a posibles variaciones.

Se podría llegar a mejorar buscando alguna forma de dar más aleatoriedad al proceso para dar pie a generar soluciones más ricas musicalmente hablando.

Por otro lado, demuestra, en otra forma musical diferente a las Corales de Bach, la fuerte estructuración que existe en las composiciones clásicas.

Se ha descartado el uso de una codificación en árbol debido a que las Corales de Bach no se estructuran en partes. Quizás para alguna extensión futura en la que intervengan modulaciones o se intente extender el algoritmo a otras formas compositivas como la Fuga, podría ser útil apoyar el algoritmo nutriéndose de codificaciones en árbol.

2.2.2. | Harmonic Navigator.

Este proyecto lo han desarrollado Bill Manaris y David Johnson del departamento de Ciencia de la Computación y Yiorgos Vassilandonakis, del departamento de música; ambos de la escuela de Charleston, USA.

Es uno de los múltiples proyectos reunidos en el libro “Evolutionary and Biologically Inspired Music, Sound, Art and Design” donde se resumen las conferencias del evento “EvoMUSART 2004” en Granada, España. [13]

Harmonic Navigator consiste en un sistema en tiempo real cuyo objetivo es explorar el espacio armónico en Corales de Bach. Para ello, a través de un *corpus* de corales, investiga a cerca de las probabilidades de las diferentes transiciones armónicas dentro de una coral.

También se va a mencionar que su interfaz es llevaba a cabo con “Kinect Engine” aunque no se va a hacer más hincapié en este detalle, puesto que no es el campo de estudio de este proyecto de investigación.

El funcionamiento general consiste en, sobre una tonalidad, el usuario va escogiendo acordes entre varias propuestas generadas por un algoritmo genético.

Este algoritmo se sirve de una base de datos de 371 corales, a las que se les suma 14.694 obras clásicas y 500 canciones de Jazz, Rock, Country y Pop; todas ellas en formato MIDI.

El insertar piezas de diferentes estilos, se explica que es para investigar la posible relación entre estos con las formas clásicas y creen que puede llegar a nutrir de más variedad musical las recomendaciones que el software hace al usuario.

Para extraer las armonías de cada una de las piezas de la base de datos, generan un sistema probabilístico, primero eliminando todas las notas de duración demasiado corta respecto a la duración media de una nota en cada una de las piezas y, posteriormente, normalizan las melodías, calculando la clave de la obra y posteriormente el tono; se selecciona las dos octavas más habituales y

respecto al tono base encontrado se elaboran el resto de grados, es decir, las progresiones armónicas.

De cada pieza se guardan, únicamente, los acordes de las progresiones en números que indican la distancia (por encima: positivos, por debajo: negativos), en número de notas, entre el acorde y el tono.

Mediante Modelos de Markov se calcula una matriz que indica con qué frecuencia se llega a una armonía a partir de cierta progresión armónica.

Para guiar al usuario en la armonía se usa el algoritmo genético que se sirve de una función de fitness que se basa en “power-law”. Esto genera un espacio armónico prácticamente infinito. Las armonías seleccionadas por el algoritmo se basan en el entorno de la composición.

El fitnes basado en “power-law” consiste en la ley de Zipf para la recuperación de información musical [14]. Consiste en un modelo estadístico basado en la probabilidad de que un evento ocurra, es decir, que una palabra aparezca en un libro es proporcional a la frecuencia con la que es encontrada:

$$P(f) = 1/\text{frecuencia}$$

En este fitness se ha encontrado similitudes entre esta ley y la música para llegar a calcular distancias de acordes, densidades de grados y los propios acordes que aparecen en una armonía.

Por otro lado, la población inicial es creada con el modelo de Markov y cada individuo codifica una progresión armónica. Las mutaciones son aleatorias seleccionando un acorde del individuo y cambiándolo mediante el modelo de Markov. Finalmente, los cruces son realizados en un único punto seleccionado de forma aleatoria con dos individuos.

Se concluye que los resultados obtenidos son excelentes. [13]

Este software es muy representativo del concepto explicado en la introducción de este documento, donde se defiende que las Corales de Bach están fuertemente estructuradas.

Por otro lado, los desarrolladores abarcan el problema desde un *corpus*, idea que, en principio, no es interesante en esta investigación puesto que se busca armonizar una línea de bajo o soprano igual que lo haría un alumno y sin la necesidad de servirse de la ayuda de un humano en la toma de decisiones de la progresión armónica, si no que se encara respecto a las reglas de la forma musical.

El usar un fitness basado en una ley probabilística que se nutre de la frecuencia de aparición dentro de las piezas de la base de datos, resulta demasiado enrevesado y con cierta incoherencia para la investigación que compete a este documento, ya que no tiene sentido buscar progresiones armónicas frecuentes en otros estilos de composición, puesto que las Corales

de Bach permiten unas progresiones muy determinadas y muy poco coloridas, no se cree que pueda beneficiarse de coloreados armónicos exteriores.

Respecto al algoritmo, se comprueba que sigue las directrices del GenJam mediante cruces simples y, como diferencia, sus mutaciones son sobre cada gen y de forma aleatoria y por sustitución.

Este algoritmo genera una gran limitación ya que, aunque en las corales de Bach sea necesario controlar la armonía de la progresión de acordes, se requiere de una amplia libertad para que cada una de las cuatro voces se mueva de forma independiente.

2.2.3. | Creation of Music Chord Progression Suited for User's Feelings Based on Interactive Genetic Algorithm.

Este proyecto ha sido desarrollado por Makoto Fukumoto del departamento de Ciencia e Ingeniería de la Computación del Instituto Tecnológico de Fukuoka, Japón.

En este trabajo el objetivo es servirse de un algoritmo genético interactivo para generar progresiones de acordes que expresen ciertos sentimientos pedidos por un usuario.

Para esto, se utiliza una limitación de 24 acordes: 12 mayores a partir de DoM y 12 menores y todos los acordes tienen el mismo tamaño rítmico.

Para el fitness se utiliza a 14 personas en dos experimentos:

- En el primer experimento se realizan 13 generaciones de 8 individuos, en las que cada persona tiene que evaluar la progresión de acordes según sean más oscuros o más brillantes. Se utiliza la ruleta como método de selección, cruce de un punto aleatorio y un 10% de probabilidad de mutación.

- En el segundo experimento se demuestra la eficacia de los resultados, generados en el anterior proceso, mediante la comparación de las progresiones más representativas generadas. Estas progresiones son escuchadas libremente por los sujetos para evaluar de 0 a 7 su calidad.

Los resultados que presentan son muy buenos, viendo una gran mejora en el proceso evolutivo y observando que si se realizan más generaciones aún se podría conseguir más mejora; esto se puede comprobar en su documentación [15]

La idea de generar las progresiones por medio de un fitness realizado por sujetos humanos es, por un lado, muy útil a la hora de encontrar una valoración sobre opiniones subjetivas; pero, por otro lado, limita mucho la dimensión de las

poblaciones y el número de generaciones; puesto que el tiempo invertido en la experimentación sería muy elevado con dimensiones y número de generaciones alto.

En este proyecto, en contraposición a lo analizado en el estado del arte, se busca demostrar la existencia de reglas objetivas que permiten generar un fitness para las progresiones. Se ha usado interactividad, pero de forma más eficiente respecto al tiempo, buscando que un experto compositor indique, además de las reglas utilizadas en las Corales de Bach, aquellos rasgos "subjetivos" que se pueden racionalizar y evaluar numéricamente.

2.3. | Extrayendo conclusiones.

Una vez analizado todo el estado del arte y comentado todos los proyectos que se ha creído conveniente destacar para nutrir el análisis y las soluciones encontradas al problema planteado, se podrían sacar ciertas conclusiones:

Por un lado, observando y analizando, en general, el campo de estudio en el que esta investigación se encuentra, se puede concretar que tiene mucha actividad en las últimas décadas aunque se observa un claro techo que mantiene todas las investigaciones dando vueltas entorno a perspectivas similares.

Es por esto, que con este trabajo se intenta abrir un camino que comienza por estudiar la generación de composiciones musicales desde un punto de vista puramente objetivo y evaluable de forma matemática, nutriéndose de las formas musicales clásicas más estructuradas y regladas utilizadas en la enseñanza musical.

De esta forma, se cree que puede llegar a encontrarse, en futuras investigaciones, algún punto clave que habrá nuevas puertas en este campo de investigación.

Por otro lado, el análisis más enfocado a los algoritmos genéticos y a los diseños comunes, sugieren el uso de sistemas simples con bases de datos o en su defecto fitness interactivos.

Por ello, se va a procurar investigar entorno a estos algoritmos genéticos con mutaciones sencillas y aleatorias, selecciones por torneo y cruces simples, pero enfocando la interactividad como ya se ha comentado anteriormente, desde la objetividad de las decisiones de un maestro de la composición a la hora de evaluar los ejercicios de sus alumnos, codificándolas y resumiéndolas en reglas matemáticas para usarlas como parte del fitness del algoritmo.

03. | ANÁLISIS DEL PROBLEMA.

A lo largo de la historia de la música, los grandes nombres han aportado formas musicales muy regladas y con estructuras rigurosas como herramientas en la enseñanza de las habilidades de composición.

Algunas de estas formas son la Fuga de Escuela o las Corales de Bach.

El estilo compositivo de estas formas musicales, otorga a los aprendices amplias habilidades para reconocer distancias, progresiones armónicas, errores en la armonía, etc. Además, les otorga gran fluidez a la hora de componer y analizar las obras, ya que dominan las formas musicales más estrictas.

Respecto al campo de la inteligencia artificial (IA), uno de los techos que se busca romper es poder crear arte de forma totalmente automática y subjetiva, pues se cree que en ello reside una de claves para conseguir entender el funcionamiento de nuestro cerebro y lograr una IA que replique dicho funcionamiento.

Durante las últimas décadas, como se ha mostrado en el análisis del Estado del Arte, se han aportado numerosos proyectos de investigación y de herramientas en el campo de la generación automática de arte. Una de las ramas de este campo es la música evolutiva.

Este tipo de música es generada mediante computación evolutiva; usando algoritmos genéticos o estrategias evolutivas y, en gran número de ocasiones, con apoyo de Modelos de Markov, grámaticas, Redes de Neuronas, interactividad con sujetos humanos, etc.

En todos los análisis se enfrenta la creación de música desde la perspectiva de la necesidad de no poder reducir a código las decisiones, comúnmente entendidas como subjetivas, de los compositores. Por esto, se hacen uso de personas que evalúen los resultados, redes de neuronas entrenadas con inmensas bases de datos de música, etc.

La hipótesis que se plantea en este proyecto es si puede existir una línea de investigación que se enfoque en reducir, lo máximo posible, esa subjetividad compositiva a conceptos lógicos evaluables por una máquina.

Para ello, estudiando la historia de la enseñanza musical, que se ha comentado anteriormente; se pretende buscar algoritmos que aprendan de la misma forma que aprenden los alumnos en las diferentes asignaturas de composición. Es decir, afrontar esta línea de investigación desde el lado de los conocimientos musicales; poniendo, al servicio de los mismos, las herramientas de computación de las que se dispone actualmente.

En otras palabras, centrar la perspectiva en entender los procesos mentales de un compositor y, a raíz de los conocimientos encontrados, usar las técnicas de computación para replicar la lógica que se encuentra debajo de éstos.

En este proyecto, se da un primer paso en este objetivo planteado. Usando la forma compositiva que primero aprende un compositor, las Corales de Bach. Se plantea generar un algoritmo genético que busque soluciones viables sobre los ejercicios, basados en esta forma musical, a los que los alumnos se enfrentan en la asignatura de armonía.

Se intenta demostrar con este estudio, que el algoritmo genético, sin más apoyo que un fitness basado en la codificación de las reglas musicales y la lógica que reside en las decisiones de un maestro de composición, a la hora de corregir dichos ejercicios, es capaz de dar resultados satisfactorios.

De forma resumida, el algoritmo debe armonizar una línea melódica dada, que haga referencia a un bajo o a la voz soprano (tiple) de una Coral de Bach.

De forma gráfica, el objetivo es obtener lo siguiente:

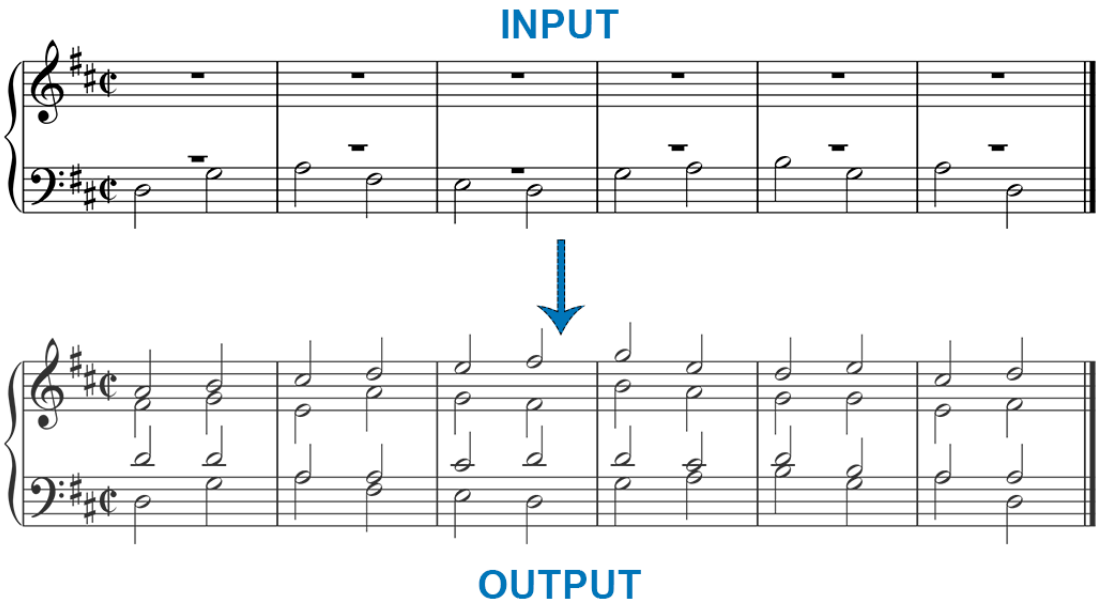


Figura 9 - Input/output

3.1. | Posibles soluciones.

Para analizar las posibles soluciones del problema planteado, se divide el problema en tres partes:

- **Codificación de las soluciones:** Cómo traducir toda la información que contiene una Coral de Bach a un lenguaje binario para crear los individuos de las poblaciones generadas por el algoritmo genético.
- **Fitness:** Cómo codificar todas las reglas formales y estéticas que se deben tener en cuenta para que una Coral de Bach este compuesta de forma adecuada.
- **Algoritmo Genético:** Cuál es el diseño a usar para este algoritmo; tipo de cruces, técnica de selección y tipo de mutaciones.

3.1.1. | Codificación de las soluciones.

En una Coral de Bach es necesario tener en cuenta mucha información sobre intervalos entre notas, tanto de forma melódica como armónica, tipos de acorde, etc.

Por esta razón, es necesario que la codificación de las soluciones esté apoyada por estructuras de datos que identifiquen tota la información posible dentro de esta codificación.

Para obtener una visión general del análisis de la información que es necesaria conocer de esta forma musical, y las necesidades que esto genera en la codificación; se presenta una tabla similar a las generadas para los requisitos de un sistema, con el fin de dar una visión gráfica que englobe todos estos conocimientos y asegurar la eficacia de la codificación en su diseño final.

Id	Tipo de Información.	Descripción.	Necesidades en la codificación.
00	Nombre de cada nota.	Es necesario conocer cuál es el nombre de la nota para situarla en el espacio armónico y melódico.	7 nombres de notas.
01	Alteración de cada nota.	Una nota puede ser natural, sostenido (si esta un semitono por encima), o bemol (si está un semitono por debajo)	3 alteraciones.
02	Octava de cada nota.	En las Corales de Bach se necesita poder contemplar 4 octavas diferentes.	4 octavas.

Id	Tipo de Información.	Descripción.	Necesidades en la codificación.
03	Tonalidad.	La tonalidad permite poder contextualizar el resto de la información respecto a las reglas.	La tonalidad es un dato proporcionado en la entrada junto con la línea melódica.
04	Intervalos armónicos.	Las distancias en tonos entre las notas de cada uno de los acordes de la composición. Sin tener en cuenta la octava.	Se calcula conociendo la nota, la alteración y la octava.
05	Información de cada acorde.	De los acordes se necesita conocer su grado en la tonalidad, su inversión, el tipo de acorde, las notas repetidas, si está dentro de la tonalidad de la obra,	Se obtiene con los intervalos armónicos.
06	Intervalos melódicos.	Distancia en tonos entre las notas de cada una de las voces. Entre pares. Sin tener en cuenta la octava.	Se calcula conociendo la nota, la alteración y la octava.
07	Distancias melódicas.	Distancia en número de notas entre las notas de cada una de las voces. Entre pares. Teniendo en cuenta la octava y sin tenerla en cuenta.	Se calcula conociendo la nota, la alteración y la octava.
08	Distancias armónicas.	Las distancias en tonos entre las notas de cada uno de los acordes de la composición. Teniendo en cuenta la octava y sin tenerla en cuenta.	Se calcula conociendo la nota, la alteración y la octava.
09	Dirección del intervalo/distancia	Conocer si el intervalo o la distancia entre dos notas es ascendente o descendente, tanto de forma armónica como melódica	Se calcula conociendo la nota, la alteración y la octava.

Id	Tipo de Información.	Descripción.	Necesidades en la codificación.
10	Pertenencia de cada nota a la tessitura de la voz a la que pertenece.	Para cada una de las notas en cada voz es necesario que no superen unos límites de graves y agudos.	Se calcula conociendo la nota, la alteración y la octava.

tabla 1 - Información de la composición.

Una vez se ha analizado toda la información que se necesita conocer de una Coral de Bach, se puede observar que la codificación necesaria debe tener en cuenta el nombre de la nota (7 posibilidades), el nombre de la alteración (3 posibilidades) y la octava a la que pertenece dicha nota (4 posibilidades). Y se conoce que las composiciones están formadas de cuatro voces, es decir, cada acorde contiene cuatro notas compuestas de nombre, alteración y octava.

Por otro lado, se observa que las Corales de Bach utilizan siempre una figura rítmica estable, pueden duplicar la duración de una nota en el caso de generar una cadencia, pero no influye a la hora del proceso compositivo. Por lo tanto, se cree que no es necesario que la codificación contemple el ritmo de cada nota.

Teniendo en cuenta el análisis realizado y las conclusiones extraídas en el proceso del estado del arte, se encuentran varias hipótesis de codificación:

- Se puede codificar cada uno de los conceptos por separado (nombre, alteración y octava). Para codificar las siete posibilidades de nombres (do, re, mi, fa, sol, la y sí), se necesitarían 3 bits, 2 bits para las tres alteraciones (#, b y natural), y 2 bits para las cuatro octavas (1, 2, 3, 4); un total de 7 bits cada nota y 28 bits para cada acorde. Esta codificación genera ciertos problemas de redundancia en las codificaciones del nombre y la alteración, al codificar 8 y 4 posibilidades pero solo necesitar 7 y 3.

  Do # 1 = 001(Do) 10(#) 00(1)

  Re b 1 = 010(Re) 11(b) 00(1)

- Otra posible codificación sería entender las notas como un todo, es decir, se dispone de 7 tipos de nombres de notas cada uno con 3 posibles alteraciones, que darían 21 notas; por otro lado, cada una de éstas puede estar en 4 octavas diferentes, es decir, 84 notas. Para codificarlo, no valdría con 6 bits ya que codificarían sólo 64 posibilidades sino que se necesitarían 7 bits, 128 posibilidades. Es decir, se utilizaría el mismo número de bits que en la codificación anterior pero con redundancia en 44 codificaciones vacías y una peor manipulación de la información.

Do b 1 = 0000000

Do 1 = 00000001

Do # 1 = 0000010

Re b 1 = 0000011

- Si se intentase entender la nota como nombre y alteración y por separado su octava; se tendría una codificación para 21 notas (5 bits) y una codificación para 4 octavas (2 bits). De esta forma se generan una redundancia en 11 codificaciones vacías.

Do b 1 = 00000(Dob) 00(1)

Do 1 = 00001(Do) 00(1)

Do # 1 = 00010(Do#) 00(1)

Re b 1 = 00011(Reb) 00(1)

- Por otro lado, se puede restringir el número de notas posibles a las 7 de cada tonalidad Mayor u 8 en las tonalidades menores. Pero esto saldría fuera de uno de los objetivos de la investigación referente a experimentar como el algoritmo se comporta teniendo a su disposición todas las notas, ya que esto le permite encontrar soluciones más variadas.

- Por último, se podría reducir el número de notas a los sonidos disponibles, es decir: Do# es el mismo sonido que Reb. De esta forma, en la segunda codificación se dispondría de 12 notas por 4 octavas, es decir 48 notas. Serían necesarios 6 bits (64 posibles combinaciones), es decir, una redundancia de 16 codificaciones vacías. A esto se le suma la pérdida del enfoque que se le quiere dar al algoritmo, queriendo que resuelva el problema desde la visión de un músico y éste entiende de forma diferente el concepto de Do# y Reb.

3.1.2. | Fitness.

La función de fitness es el concepto clave de este proyecto. Se encarga de evaluar de forma numérica cada una de las soluciones generadas por el algoritmo genético.

En la mayoría de casos analizados en el Estado del Arte se encuentran fitness interactivos en los que se necesita de humanos que evalúen cada una de los individuos creados; esto reduce significativamente las grandes capacidades de los algoritmos genéticos, limitando en gran medida el número de generaciones y la dimensión de las poblaciones.

Por otro lado, también interfiere en uno de los objetivos del proyecto que es buscar la manera de sintetizar el conocimiento subjetivo que aporta la interactividad.

Para solucionar esto, los proyectos analizados se sirven del apoyo de gramáticas, redes de neuronas entrenadas con bases de datos inmensas, etc.

Pero si se afronta el problema desde la perspectiva del aprendizaje musical, un alumno no tiene un bagaje en la composición y su única forma de generar soluciones es a partir de lo que le enseñan sus maestros y las explicaciones que les dan de los detalles estéticos (subjetivos).

Es por esto que se plantea la hipótesis de crear una función fitness que se nutra de todas las reglas de las Corales de Bach (extraídas de los apuntes del autor del proyecto y del libro de armonía de Zamacois [4]) y, a su vez, realizar una entrevista con un maestro de composición de un conservatorio profesional de música que detalle aún más el fitness y añada nuevas reglas estéticas que emplea para enseñar a sus alumnos.

Como último paso en el análisis del concepto del fitness, se muestra en una tabla similar a las empleadas en requisitos, que relacione las diferentes reglas de las Corales de Bach con la información codificada contenida en la tabla anterior tabla.

Id	Regla	Excepciones	Id: Tipo de Información
F01	Prohibidos los cruces entre voces.	Sin excepciones.	08, 09
F02	Prohibidos acordes fuera de la tonalidad	Sin excepciones	05
F03	Prohibidas distancias superiores a la 8ª entre las tres voces más agudas.	No se tiene en cuenta el bajo.	08
F04	No se permiten 5ª, 8ª o unísonos paralelas.	5ª si la segunda no es mayor y las voces no son externas.	04, 08
F05	Se permiten 5ª, 8ª o unísonos directas	En voces externas y la soprano no viene de un intervalo de 2ª.	07, 08
F06	Se prohíben los intervalos melódicos aumentados.	Sin excepción.	06

Id	Regla	Excepciones	Id: Tipo de Información
F07	Se prohíben los intervalos de 7ª, 9ª o superiores.	Sin excepción	06
F08	No se permiten que todas las voces realicen la misma dirección de intervalo en un acorde.	Sin excepción	09
F09	El primer y último acorde debe ser el I en estado fundamental.	Sin excepción	05
F10	Se debe terminar con los acordes V y I o I en segunda inversión V en fundamental y I en fundamental.	Sin excepción	05
F11	No se puede duplicar la sensible en los acordes V y VII	Sin excepción	05
F12	No se puede duplicar la séptima, ni la novena del acorde.	Sin excepción	05
F13	La sensible se altera un semitono por encima en las tonalidades menores.	Sólo en los acordes V y VII	00, 01, 03, 05
F14	La sensible debe realizar un intervalo de 2ª ascendente.	Sólo en los acordes V y VII	05. 07, 09
F15	La séptima y la novena deben realizar un intervalo de 2ª descendente.	Sin excepción.	05. 07, 09
F16	Una voz no puede cruzarse con otra del acorde anterior.	Sin excepción	07, 09

Id	Regla	Excepciones	Id: Tipo de Información
F17	Se debe cumplir las progresiones armónicas estipuladas por norma.	Sin excepción.	05
F18	Se prohíbe el tritono.	Sin excepción.	06
F19	No se permite sobrepasar tessitura de cada voz.	Sin excepciones.	10

tabla 2 - Reglas Corales de Bach Parte 1

Es necesario mencionar que las progresiones armónicas que se deben cumplir son (6 indica la primera inversión):

Grado	Posibles progresiones
I	TODOS
II	IV, V, VII
III	II6, IV, VI
IV	I, II, IV, I, VII
V	I, IV6, VI, VII
VI	II, IV, V, VII
VII	I, V, VI6

tabla 3 - Progresiones armónicas

3.1.3. | Algoritmo Genético.

Una vez se ha estudiado las codificaciones de los individuos y la función de fitness, queda por analizar el algoritmo genético que se podría utilizar en el proyecto.

En primer lugar, se va a tratar los posibles funcionamientos del algoritmo genético.

- Por un lado, se puede recurrir al procedimiento simple, seguido por la gran mayoría de los proyectos analizados; explicado brevemente, se genera una población inicial, se seleccionan los mejores, éstos se cruzan y los descendientes tienen cierta probabilidad de mutaciones y, con esta nueva generación, comienza el ciclo de nuevo.

- En este problema no existe una única solución válida, es por esto, que una mejora para el algoritmo se podría basar en una evolución mediante nichos, donde el algoritmo al evaluar sus individuos también tiene en cuenta cuales son los más diferentes entre los mejores.

- Llevando esto un grado más lejos, se podría crear un proceso evolutivo por islas. Consistiría en generar varias poblaciones distintas que evolucionen por separado pero que compitan entre sí por encontrar la mejor solución.

Una vez se ha analizado el posible funcionamiento del algoritmo, es necesario estudiar qué posibilidades se encuentran en los tipos de mutación, tipos de selección y tipos de cruces.

Operadores de Selección.

Por lo general, en los proyectos estudiados se encuentran los operadores de selección de tipo torneo o ruleta.

- **Selección por torneo**: Enfrenta a dos o más individuos y selecciona al mejor de ellos. Si los torneos son con individuos elegidos de forma aleatoria o no; aunque se ha demostrado que lo que más funciona es la aleatoriedad. Además, es necesario probar si es mejor realizar los torneos entre dos, tres o más individuos. Por lo general, se suelen utilizar dos o tres.

- **Selección por ruleta:** Se seleccionan los individuos de forma que los que tienen más probabilidad de ser elegidos son aquellos que tienen mejor fitness. Esto suele descompensar la variabilidad genética en los casos donde hay grandes diferencias entre los fitness.

Por lo estudiado en el Estado del Arte, suele dar mejores resultados la selección por torneo.

Por otro lado, se puede asegurar que no se pierden los mejores individuos con una selección elitista, en las que un porcentaje de los mejores individuos pasan a la siguiente generación sin cambios.

También existe la posibilidad de cambiar los criterios de selección, al igual que de fitness según avanza la evolución; pero no se han encontrado muchos casos de esta técnica y se deja para futuras investigaciones.

Tipos de cruces.

Existen muchas posibilidades de técnicas de cruce (reproducción) entre individuos. Las más comunes son:

- **El cruce simple:** Se realizan dos o más partes de cada uno de los individuos a cruzar y los descendientes son las mezclas de estas partes.
- **Cruce uniforme:** Se mezcla gen por gen de cada uno de los individuos a cruzar.
- **Otros aritméticos:** Se emplean operaciones aritméticas sobre los individuos a cruzar para generar los descendientes.

Por lo general, se ha analizado que el cruce más común es el simple, ya que los otros son para problemas muy específicos.

Finalmente, es necesario tener en cuenta el número de individuos que van a participar en el cruce, por lo general suelen ser dos.

Tipos de mutaciones.

Al analizar estas técnicas se han encontrado todo tipo de posibilidades, como se ve en el proyecto genjam [7], que no solo usa una técnica. Por otro lado, se encuentran cruces bajo plantillas, que pueden ser utilizados en ciertas ocasiones para proteger material genético.

En el caso de las mutaciones, el proyecto va a trabajar sobre las más simples, ya que no se cree necesario que influya positivamente el ir más allá.

Esto es debido, al sentido del problema, el cambio de un bit influirá en que se cambiará una nota por otra diferente y esto generara bastante repercusión en el nuevo individuo. Es por esto, que se va estudiar sobre la frecuencia de esta mutación bit a bit que normalmente ronda desde 0.2% para debajo de probabilidades que una mutación ocurra.

04. | DISEÑO DE LA SOLUCIÓN.

Una vez se han estudiado todas las posibles decisiones a tomar para el diseño y se han analizado las reglas a cumplir en la codificación y en el fitness, se procede, en este apartado a explicar las decisiones tomadas y el diseño final, argumentándolo mediante pruebas.

En primer lugar, es necesario definir el **entorno de pruebas** sobre el que se va a trabajar; seguidamente, se presenta el **diseño inicial del algoritmo genético**; a continuación, se llega al centro del proyecto que reside en **las estructuras de datos utilizadas** y la **función fitness.** Sobre la primera versión de la función fitness ya se van a ofrecer los primeros resultados de las pruebas y, una vez aplicados los nuevos conocimientos sobre el fitness, en la **entrevista** con el maestro Álvaro Israel Gómez Alvarado; se presentarán nuevas pruebas para mostrar los cambios que esto ha conllevado.

Finalmente, cuando toda esta parte del diseño está explicada, se procede a realizar pruebas sobre el algoritmo genético y sus posibles mejoras.

El entorno de pruebas permanece privado en la cuenta de GitHub del autor [16] del proyecto, pero se planeta liberarlo públicamente una vez se dé por concluida el proyecto en cuestión y se estudie si se pueden realizar mejoras que necesiten la privacidad del mismo.

4.1. | Entorno de pruebas.

Para realizar las pruebas que nutran las hipótesis planteadas sobre el diseño del algoritmo genético, el fitness y los objetivos del proyecto, se requiere implementar todo sobre un lenguaje de programación orientado a objetos; para esto, se precisa de un entorno de desarrollo del lenguaje.

Por otro lado, se precisan de programas para generar las gráficas de evolución de los resultados del algoritmo. A su vez, es necesario definir la entrada y la salida que la implementación precisa.

4.1.1. | Python 3.6.

El lenguaje sobre el que se va a trabajar es Python en su versión 3.6. [17]

Python es un lenguaje orientado a objetos que está en constante evolución desde que se presentó su primera versión en 1991. Fue desarrollado en el centro de investigación CWI (Centrum Wiskunde and Informatica) de Ámsterdan, por Guido van Rossum con el objetivo de eliminar las restricciones de otro lenguaje que usaban llamado ABC.

A partir de 2001, Python es controlado desde la fundación sin ánimo de lucro, “Python Software Foundation”.

Como anécdota mencionar que el nombre de Python viene de la serie *"Monty Python's Flying Circus"*.

Está sujeto bajo una licencia GLP (GNU General Public Licence) llamada Software Foundation Licence.

El carácter open source de Python permite que se nutra de gran cantidad de aportes de la comunidad.

Actualmente, se mantienen en paralelo las versiones 2.X y 3.X, debido al gran cambio que supone esta última respecto a la otra, y a la resistencia de los usuarios de realizar dicho cambio.

Se ha decidido implementar el entorno de pruebas en este lenguaje, debido a su juventud y la posibilidad de generar un aporte con mayor impacto en la comunidad.

También se ha encontrado en Python un lenguaje muy limpio y claro que ayuda a la comprensión del mismo y a una manipulación más sencilla y visual. Además, se conoce que es más potente que su similar, JAVA, en lo referente al tratamiento de problemas de algoritmos genéticos.

4.1.2. | Eclipse Luna y PyDev.

El entorno de pruebas se ha desarrollado sobre Windows y Linux de forma paralela, es por esto, que se ha decidido utilizar el entorno de desarrollo **Eclipse**, en su "**Luna**". [18]

Eclipse es un entorno de desarrollo de código abierto típicamente utilizado en proyectos sobre el lenguaje Java.

Originalmente, fue desarrollado por IBM, pero actualmente se encarga de él, la organización sin ánimo de lucro "Eclipse Foundation". Está sujeto bajo una licencia GNU GPL llamada Eclipse Public License.

Como se ha comentado, es un entorno de desarrollo realizado para Java en su esencia, pero extendido a PHP, C, etc.

Para poder desarrollar con el lenguaje Python sobre este entorno es necesario utilizar **PyDev,** una Python IDE (Entorno de Desarrollo Integrado) para Eclipse. [19] Este IDE es de código abierto, desarrollado entre 2003 y 2005.

4.1.3. | Microsoft Excel.

La evolución resultados de las pruebas es representada mediante gráficas generadas con el programa del paquete Microsoft Office, Excel en su versión 2013. [20]

Es un programa de pago basado en hojas de cálculo que permite realizar infinidad de tareas: contables, financieras, etc.

También va a ser aprovechado en la documentación para generar los presupuestos necesarios.

4.1.4. | Archivo de entrada y salida.

La entrada y la salida se dan, en principio, bajo ficheros de texto plano.

En la entrada el fichero contiene la tonalidad, la longitud de la obra en número de acordes, el número de la voz dada (entre 0 la más aguda y 3 la más grave). Y la línea melódica. También se ha generalizado para contemplar la posibilidad de una entrada de una coral con las 4 voces completas.

```
t: reM

i: 12 3

s:

ren3 soln3 lan3 fa#3 min3 ren3 soln3 lan3 sin3 soln3 lan3 ren3
```

Por otro lado, la salida generada proporciona las notas que componen la coral y su fitness.

```
lan2 sin2 do#1 ren1 min1 fa#1 soln1 min1 ren1 min1 do#1 ren1

fa#2 soln2 min2 lan2 soln2 fa#2 sin2 lan2 soln2 soln2 min2 fa#2

ren2 ren2 lan3 lan3 do#2 ren2 ren2 do#2 ren2 sin3 lan3 lan3

ren3 soln3 lan3 fa#3 min3 ren3 soln3 lan3 sin3 soln3 lan3 ren3

-2
```

A su vez, durante todo el proceso de evolución se muestra por pantalla el número de generación, los fitness de la población y el mejor fitness conseguido hasta el momento. Además en un fichero csv se guardan en tres columnas: el mejor fitness de la generación, el peor y el fitness medio.

Con este archivo csv y Excel se generan las gráficas de evolución de cada prueba.

4.1.5. | Sibelius, Lilypond y Abjad.

Por último, comentar que las partituras que figuran en este documento han sido generadas con el programa de edición de partituras **Sibelius**, desarrollado por la compañía Avid y sujeto bajo una licencia de pago mensual.

Comentar la posibilidad de utilizar el software **Lillypond** open source, que permite generar partituras en Latex a partir de ficheros de texto plano que se podrían implementar con un escritor en Python. [21]

Por otro lado, existe la librería open source, para Python, Abjad que permite la composición y edición de música sobre partituras. [22]

4.2. | Algoritmo Genético.

Una vez se ha estudiado las posibles codificaciones de los individuos, y el ámbito de decisiones a tomar frente al diseño del algoritmo genético, se presenta un diseño inicial del mismo, utilizado para probar las estructuras de datos y el la efectividad de combinar el fitness con el algoritmo genético.

El diseño se va a estructurar en 5 fases de desarrollo: El diseño general del algoritmo, la codificación de los individuos, el proceso de selección, el proceso de cruces y las mutaciones.

4.2.1. | Diseño general.

En esta primera versión del algoritmo genético se ha optado por realizar un diseño simple, ya que una vez realizado el fitness se podrán realizar las pruebas convenientes y estudiar posibles mejoras.

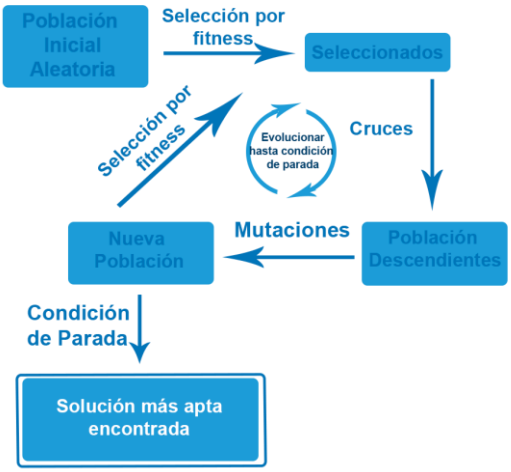


Figura 10 - Diseño general de la solución.

Este diseño consiste en generar una **población inicial aleatoria**, ya que se busca que el algoritmo empiece desde el desconocimiento total del problema. El tamaño de esta población y las siguientes se fija, inicialmente, en 10 individuos; pero, se realizarán las pruebas pertinentes para ver si dan mejores resultados otros tamaños.

Una vez se dispone de la población inicial, se realiza el proceso de selección de los más aptos respecto al fitness. De los seleccionados, se realizan los cruces pertinentes y se genera una nueva población con los descendientes. Estos nuevos individuos se ven sometidos a mutaciones bajo cierta probabilidad y esto da lugar a una nueva generación que da paso a comenzar de nuevo el ciclo evolutivo.

Eso se realiza hasta cumplir la condición de parada, en este caso se ha utilizado tanto la parada manual cuando se analiza que el algoritmo está estancado, como la parada en número determinado de generaciones para comparar unos resultados con otros en las mismas condiciones.

Una vez se cumple la condición de parada el algoritmo devuelve la solución codificada en el individuo más apto respecto a su fitness encontrado en todo el proceso evolutivo.

4.2.2. | Codificación de las soluciones.

La codificación de las soluciones, como se ha analizado, previamente tiene que contener el nombre de la nota (do, re, mi, fa, sol, la, si), la alteración (#: sostenido, b: bemol, n: natural) y la octava a la que corresponde (1 la octava más grave y 4 la más aguda).

Se entiende como la codificación con menos redundancias a solucionar y la más manipulable a la hora de utilizarla para evaluar a los individuos en la función de fitness.

No se han tenido en cuenta los dobles sostenidos, ni los dobles bemoles, porque no se usan en este tipo de composiciones.

El diseño de la codificación consiste en traducir a bits cada atributo por separado, es decir:

- **Nombre de la nota:** Se codifica con tres bits (do: 001, re: 010, mi: 011, fa: 100, sol: 101, la: 110, si: 111). La redundancia por la codificación 000 vacía se resuelve, en esta primera versión, sustituyéndola por “do”, es decir, “do” es igual a 000 y a 001; esto es debido a que es una nota muy común. Posteriormente se realizaran pruebas para usar esta codificación con una nota aleatoria o con la primera nota de la tonalidad en la que se está componiendo, ya que esta es, seguro la nota más común de esa composición.

- **Nombre de la alteración:** Se codifica con dos bits (#: 01, b: 10 y n: 11). La redundancia generada por la codificación 00 vacía, se resuelve, *a priori* como natural, es decir 00 y 11 referencian natural. Posteriormente, se probará, con una alteración aleatoria y con la alteración de la primera nota de la tonalidad en la que se está componiendo, ya que esta será la más común.

- **Octava:** Por último, se codifica la octava con 2 bits siendo 1 la octava más aguada y 4 la más grave de forma que, 1: 00, 2: 01, 3:10, 4: 11.

Cada nota será codificada con estos tres atributos de forma que un ejemplo de codificación sería:

Re b 4 = 010(Re) U 10(b) U 11(4) = 0101011 = 7bits

Cada acorde va a estar codificado con cuatro notas, es decir, por 21 bits. Pero la codificación se va a realizar por voces. Es decir, primero se codifica una ristra de bits que contenga todas las notas de una voz, luego las notas de la voz siguiente y así con todas las voces.

Finalmente, en el individuo sólo se tiene en cuenta la voz dada para evaluar el fitness, es decir, la codificación utilizada para selección, cruce y mutación no contiene la codificación de dicha voz.

Resumido de forma gráfica, el proceso de codificación es:

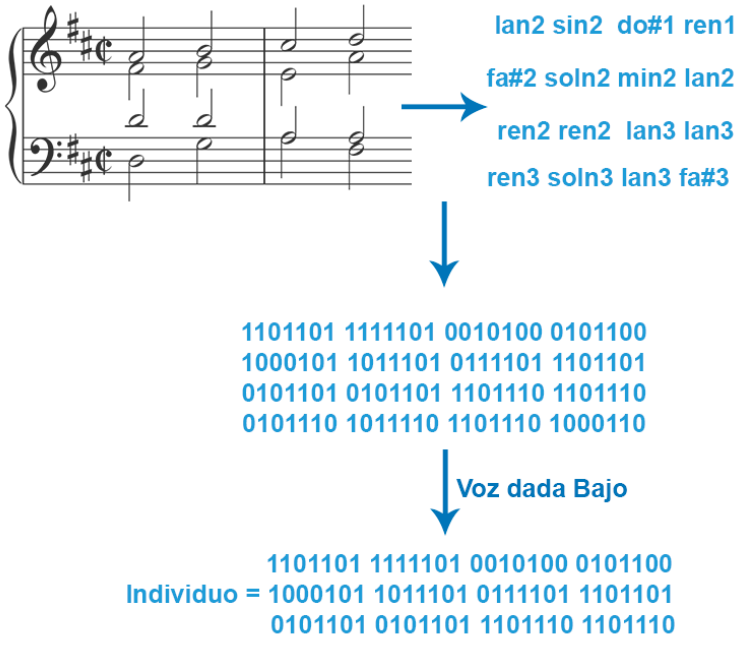


Figura 11 - Codificación

Los ejercicios, basados en Corales de Bach, suelen contener entre 15 y 20 acordes, es decir, una media de 300 bits por individuo. Esto conlleva unas 2^{300} posibles combinaciones, por ende, es una codificación que necesariamente necesita un algoritmo genético para encontrar una solución adecuada.

4.2.3. | Operadores de selección.

En el diseño de las operaciones de selección, teniendo en cuenta el análisis previo realizado, se ha optado por una **selección por torneo,** que es la más utilizada, según se ha deducido del análisis del Estado del Arte.

Esta selección se va a realizar enfrentando a 3 individuos escogidos al azar, se permite escoger al mismo individuo en diferentes enfrentamientos. De estos 3 individuos se seleccionará el que mejor fitness tenga, es decir, el mejor adaptado al problema.

La decisión de definir torneos de 3 individuos viene dada por la hipótesis de que 2 individuos tienen menos posibilidades de eliminar los individuos menos aptos que un torneo con 3.

Figura 12 - Selección

Como se ve en la Figura 12, en esta fase de selección y en el resto de fases del proceso evolutivo, se utiliza la técnica de **selección elitista** en la que el mejor individuo encontrado hasta el momento, se sustituye por el peor individuo de la siguiente población.

En el apartado mejoras sobre el algoritmo genético se realizarán las pruebas pertinentes para comprobar si la selección por torneo es mejor sobre 2, 3 o cuatro individuos.

4.2.4. | Operadores de cruce.

El tipo de operador de cruce que se ha diseñado para este algoritmo, es el que se usa de forma más general en este tipo de proyectos, es decir, el **cruce simple**; pensando en la idea de pensar en una Coral de Bach como una estructura de 3 parte: introducción, nudo y desenlace, se ha decidido realizar este tipo de cruces sobre 2 puntos en vez de 1 como es habitual.

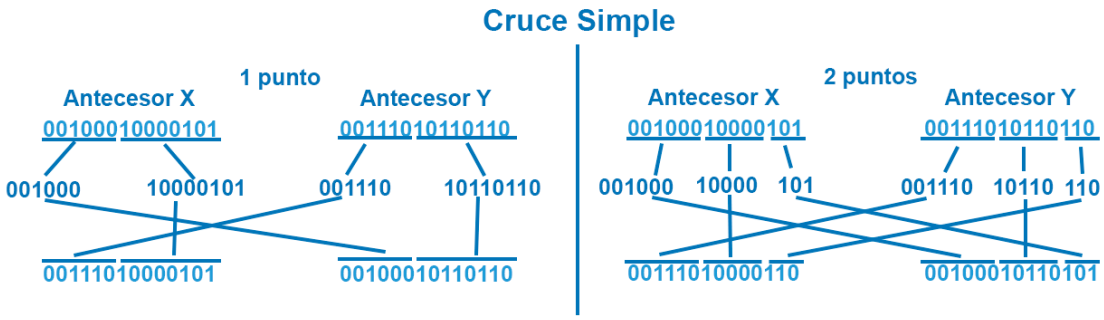


Figura 13 - Cruces

En las mejoras sobre el algoritmo genético se realizarán las pruebas tanto para un punto de cruce como para dos.

Este cruce se realiza sobre **2 individuos antecesores** (X e Y) que se dividen en tres partes mediante dos puntos seleccionados de manera aleatoria. Posteriormente, se crean **2 nuevo individuos descendientes**:

- **Descendiente 1:** Parte1 de Y + Parte2 de X + Parte3 de Y.
- **Descendiente 2:** Parte1 de X + Parte2 de Y + Parte3 de X.

Una vez generada la nueva población de descendientes se aplica, de nuevo, la selección elitista, para sustituir el peor individuo de esta nueva población, por el mejor encontrado hasta el momento.

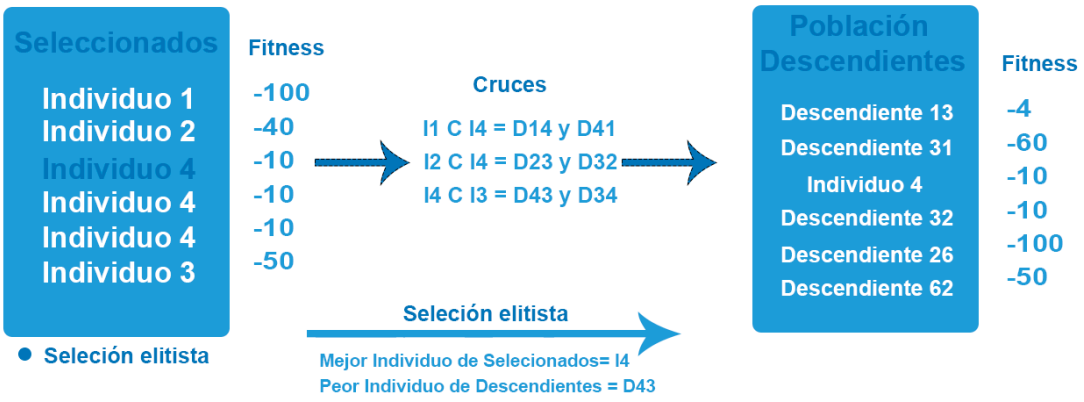


Figura 14 - Proceso de cruce.

4.2.5. | Operadores de mutación.

Como ya se ha comentado en el análisis, la mutación que se va a realizar consiste en la probabilidad de que cambie el valor de cada uno de los bits de un individuo.

Esta probabilidad se ha establecido en 0.05, pero en las mejoras sobre el algoritmo genético se realizarán pruebas sobre 0.1 y sobre probabilidades que decrezcan junto con la mejora del proceso evolutivo.

4.3. | Estructuras de datos.

Para lograr codificar todas las reglas analizadas, es necesario codificar toda la información requerida en la **tabla 1** del Análisis dela codificación.

Para cumplir este objetivo, se crean las siguientes estructuras de datos, que proporcionan dicha información a partir de la codificación de cada individuo:

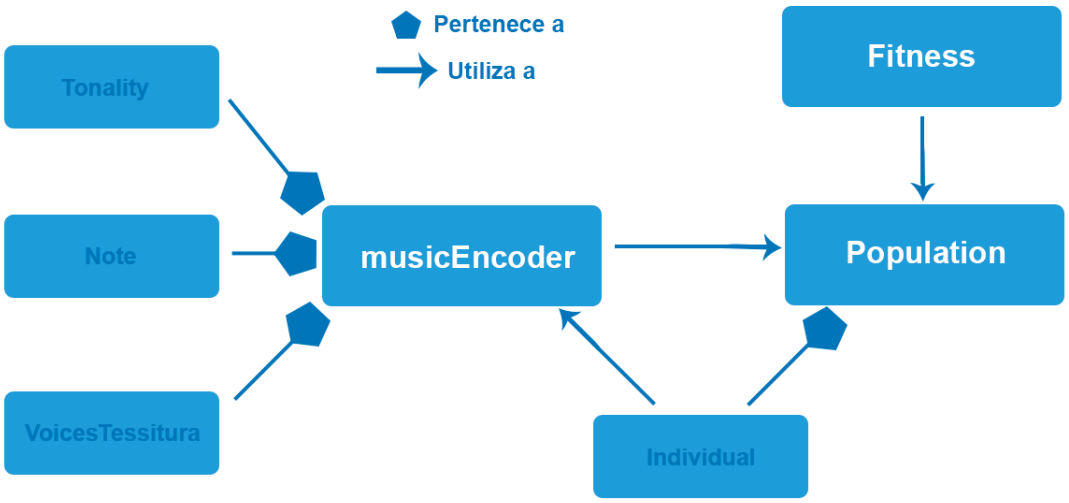


Figura 15 - Estructuras de datos

Para visualizar dónde se encuentra esta información en cada una de las partes del diagrama, se presenta una tabla, al estilo de las planteadas en los requisitos del sistema que está relacionada con la **tabla 1** del Análisis dela codificación.

Id	Tipo de Información.	Estructura de datos
00	Nombre de cada nota.	Note
01	Alteración de cada nota.	Note
02	Octava de cada nota.	Note
03	Tonalidad.	Tonality

Id	Tipo de Información.	Estructura de datos
04	Intervalos armónicos.	Individual
05	Información de cada acorde.	Individual
06	Intervalos melódicos.	Individual
07	Distancias melódicas.	Individual
08	Distancias armónicas.	Individual
09	Dirección del intervalo/distancia	Individual
10	Pertenencia de cada nota a la tessitura de la voz a la que pertenece.	VoicesTessitura, Individual

tabla 4 - Relación entre información y estructuras de datos

Una vez presentada la visión general de las estructuras de datos, se procede a explicar, en los siguientes puntos, cada una de estas estructuras.

Para ello se va a hacer uso de tablas del estilo de requisitos de sistema y una pequeña matriz de trazabilidad para relacionar atributos y métodos de cada clase, y su relación entre ellos.

Para cada uno de los métodos y atributos se han realizado las pruebas correspondientes para asegurar su completo funcionamiento.

4.3.1. | Tonality.

En esta estructura se encuentra toda la información relativa al campo de las tonalidades. De esta forma, al iniciar el algoritmo, se hará uso de Tonality para establecer la tonalidad de la composición y la información derivada de la misma.

Tonality presenta los siguientes atributos:

Id	Atributo	Función	Necesita
TA00	Name	Presenta el nombre de la tonalidad. Ej: Reb	-
TA01	Type	Presenta el tipo de tonalidad. Mayor (M), menor (m).	-
TA02	Relative	Presenta la tonalidad relativa.	TM05
TA03	Scale	Presenta la escala de las	TM00

Id	Atributo	Función	Necesita
		7 notas que contiene la tonalidad (cada nota con el nombre y la alteración correspondiente).	
TA04	ScaleMinor	Presenta la escala, en el caso de que la tonalidad sea menor, con la sensible aumentada un semitono.	TM00

tabla 5 - Tonality: Atributos

Tonality presenta los siguientes métodos:

Id	Método	Parámetros	Tipo	Función	Necesita
TM00	ScaleTonality	-	Objeto	Genera los atributos de scale y scaleTonality	-
TM01	validateTonality	Nombre de la tonalidad, tipo de tonalidad	Clase	Valida que la tonalidad exista.	NM12
TM02	allTonality	-	Clase	Presenta todas las posibles tonalidades.	TM03, TM04
TM03	sharpTonality	-	Clase	Presenta las tonalidades existentes con sostenidos.	-
TM04	flatTonality	-	Clase	Presenta las tonalidades existentes con bemoles.	-
TM05	relativeTonality	-	Objeto	Genera el atributo relative.	-
TM06	__init__	Nombre de la tonalidad y tipo.	Constructor	Crea una nueva tonalidad	TM00, TM05

tabla 6 - Tonality: Métodos

4.3.2. | Note.

En esta estructura se encuentra toda la información relativa a las notas.

Note presenta los siguientes atributos:

Id	Atributo	Función	Necesita
NA00	_notes	Presenta el nombre de las 7 notas.	-
NA01	Id	Presenta la codificación en bits de la nota.	-
NA012	Name	Presenta el nombre de la nota.	-
NA03	Alteration	Presenta el tipo de alteración de la nota.	-
NA04	Octave	Presenta la octava a la que pertenece la nota.	-

tabla 7 - Note: Atributos.

Note presenta los siguientes métodos:

Id	Método	Parámetros	Tipo	Función	Necesita
NM00	__init__	Id de la nota	Constructor	Crea una nueva nota a partir de su id	NM02, NM04, NM05, MN13
NM01	New	Nombre, alteración y octava de la nota.	Constructor	Crea una nueva nota a partir de su nombre, alteración y octava	NM00, NM02, NM04, NM05
NM02	nameNotesLatin	-	Clase	Presenta los nombres de las notas en formato latino (Do, re, etc) y su codificación	-
NM03	nameNotesAmerican	-	Clase	Presenta los nombres de las notas en formato americano (C, D, etc) y su codificación	-
NM04	nameAlterations	-	Clase	Presenta el nombre de las alteraciones y su codificación	-
NM05	octaves	-	Clase	Presenta las octavas y su codificación	-

Id	Método	Parámetros	Tipo	Función	Necesita
NM06	allNotes	-	Clase	Presenta Todas las notas (nombre y alteración)	-
NM07	intervalNotes	Dos ids de notas.	Clase	Calcula la distancia en tonos entre dos notas, sin tener en cuenta la octava.	NM06, NM11, NM12
NM08	intervalDirection	Dos ids de notas.	Clase	Indica la dirección del intervalo, descendente (-1), ascendente (1).	NM11
NM09	distanceNotes	Dos ids de notas.	Clase	Indica la distancia en número de notas entre dos de ellas, teniendo en cuenta la octava.	NA00, NM11
NM10	Distance	Dos ids de notas.	Clase	Indica la distancia en número de notas entre dos de ellas, sin tener en cuenta la octava.	NA00, NM11
NM11	sortNotes	Dos ids de notas.	Clase	Devuelve las dos notas ordenada, primero la más grave.	NA00
NM12	Equivalence	Nota	Clase	Devuelve la nota equivalente de igual sonido.	-
NM13	validateID	Id de nota	Clase	Devuelve true si la codificación de una nota es correcta.	-
NM14	noteRandom	-	Clase	Genera una nota Aleatoria	NM13

tabla 8 - Note: Métodos.

4.3.4. | VoicesTessitura.

En esta estructura se codifica toda la información referente a la tesitura de cada una de las voces.

Únicamente contiene los siguientes métodos:

Id	Método	Parámetros	Tipo	Función	Necesita
VM00	Soprano	-	Clase	Presenta los límites de la voz soprano	-
VM01	Contralto	-	Clase	Presenta los límites de la voz contralto	-
VM02	Tenor	-	Clase	Presenta los límites de la voz tenor	-
VM03	Bass	-	Clase	Presenta los límites de la voz bajo	-
VM04	noteInSoprano	Id de nota	Clase	Devuelve True si la nota está dentro de la tessitura de soprano.	VM00, NM08
VM05	noteInContralto	Id de nota	Clase	Devuelve True si la nota está dentro de la tessitura de contralto.	VM01, NM08
VM06	noteInTenor	Id de nota	Clase	Devuelve True si la nota está dentro de la tessitura de tenor.	VM02, NM08
VM07	noteInBass	Id de nota	Clase	Devuelve True si la nota está dentro de la tessitura de bajo.	VM03, NM08

tabla 9 - VoicesTessitura: Métodos.

4.3.5. | MusicEncoder.

Esta estructura agrupa las estructuras: tonality, note y voicesTessitura. Además implementa el método:

Id	Método	Parámetros	Tipo	Función	Necesita
EM00	voiceRandom	-	Clase	Genera una voz de tantas notas como tamaño sea la composición.	IA01, NM14

tabla 10 - MusicEncoder: Métodos.

4.3.6. | Individual.

La estructura de datos que recibe el nombre de individual es uno de los puntos centrales a la hora de la realización del fitness, ya que contiene toda la información necesaria en lo referente a la codificación de los individuos y a las estructuras de datos que almacenan la información de todos los intervalos melódicos y armónicos, acordes, tesituras, etc.

Es por esto que, en primer lugar, se va a explicar de forma detallada estas estructuras ya que su diseño es el molde para todo el fitness.

- **Score:** Es una lista que contiene cuatro listas que referencian las 4 voces. En estas listas figuran las codificaciones (ids) de todas las notas de la composición.

```
[['1101001', '0111101', '0111110'],
 ['1110110', '1110100', '1011011'],
 ['1010101', '0011000', '1000110'],
 ['0011110', '1011111', '1101111']]
```

- **Id:** Representa la codificación del individuo utilizada en el algoritmo, es decir, sin la voz dada. En este ejemplo, sin la cuarta voz:

110100101111010111110111011011101001011011101010100110001000110

- **Tessitura Table:** En esta es una lista compuesta por otras 4 listas, que hacen referencia a cada una de las 4 voces. Contiene, para cada una de las notas, si éstas están dentro de la tessitura de su respectiva voz (True) o no (False).

```
[[True, True, False]
[True, False, False]
 [True, False, True]
 [True, True, True]]
```

- **Tablas de Distancias:** Consiste en tres tablas (bassDistanceTable, tenorDistanceTable y contraltoDistanceTable) que se codifican bajo una lista compuesta de otras listas, una para cada voz superior a la que hace referencia la tabla, es decir: en el caso del bajo, una lista para tenor, otra para contralto y otra para soprano; en el caso del tener, una lista para contralto y otra para soprano; y en el caso del contralto, una única lista para el soprano. Cada una de estas listas hacen referencia a las distancias armónicas (tanto teniendo en cuenta la octava, como sin tenerla en cuenta) entre la voz en cuestión y el resto de voces superiores.

bassDistanceTable

```
[[6, 6], [20, 6], [5, 5]]
[[7, 7], [24, 3], [2, 7]]
[[5, 5], [25, 4], [6, 6]]
```

tenorDistanceTable

```
[[2, 2], [6, 3], [2, 7]]
[[6, 3], [7, 7], [7, 2]]
```

contraltoDistanceTable

```
[[7, 7], [5, 4], [6, 6]]
```

- **Tablas de intervalos:** Consiste en tres tablas (bassIntervalTable, tenorIntervalTable y contraltoIntervalTable) que se codifican bajo una lista compuesta de otras listas, una para cada voz superior a la que hace referencia la tabla, es decir: en el caso del bajo, una lista para tenor, otra para contralto y otra para soprano; en el caso del tener, una lista para contralto y otra para soprano; y en el caso del contralto, una única lista para el soprano. Cada una de estas listas hacen referencia a los intervalos armónicos (intervalo y dirección del mismo) entre la voz en cuestión y el resto de voces superiores.

bassIntervalTable

```
[[4.0, 1], [4.5, 1], [3.5, 1]]
[[0, 1], [2.5, 1], [1.5, -1]]
[[4.0, 1], [2.0, 1], [4.5, 1]]
```

tenorIntervalTable

```
[[0, 1], [3.5, -1], [1.0, -1]]
[[4.0, -1], [0.5, 1], [0, -1]]
```

contraltoIntervalTable

```
[[4.0, 1], [4.0, -1], [5.0, 1]]
```

- **Melodic Table:** Tabla codificada bajo una lista que almacena otras cuatro listas, una para cada voz, que contienen la información de los intervalos melódicos (distancia teniendo en cuenta la octava, intervalo y dirección del intervalo) para cada par de notas de la composición.

```
[[4, 2.0, -1], [8, 0, -1]]
[[15, 0, 1], [24, 3.0, -1]]
[[4, 1.5, 1], [19, 2.5, -1]]
[[4, 2.5, -1], [2, 1.0, 1]]
```

- **Chords Table:** Esta última tabla está formada por una lista que contiene la información de cada uno de los acordes de la composición (grado del acorde (desde 0: I, hasta 6: VII), tipo de acorde (triada: 3, séptima: 7, novena: 9), inversión (0: fundamental, 1: primera inversión, 2: segunda inversión, 3: tercera inversión), las notas que se repite respecto a al bajo del acorde (1 si es el propio bajo) y si el acorde pertenece a la tonalidad (true si es así, false si no)). Si el acorde no concuerda en alguno de sus datos se sustituye por “None” (término que utiliza Python para indicar vacío).
  El ejemplo se corresponde con una composición de dos acordes: uno de II grado, triada en estado fundamental, repitiéndose la quinta del acorde y

sin pertenecer a la tonalidad; el otro es un V grado, triada, en estado fundamental repitiendo la nota del bajo que es la tónica del acorde y sin pertenecer a la tonalidad.

```
[[1, 3, 0, 5, 5, False], [4, 3, 0, 1, False]]
```

Una vez explicado las estructuras que contiene, se muestra la tabla de los atributos:

Id	Atributo	Función	Necesita
IA00	_GivenVoice	Almacena el número de la voz dada.	-
IA01	_lenScore	Almacena la longitud, en número de acordes, de la composición.	-
IA02	Score	Almacena la codificación de la composición.	-
IA03	Id	Almacena la codificación de la composición sin la voz dada.	-
IA04	TessitureTable	Almacena para cada nota si pertenece a la tessitura de su voz o no.	IM03
IA05	bassDistanceTable	Almacena las distancias armónicas desde el bajo hasta el tenor, el contralto y el soprano.	IM04
IA06	tenorDistanceTable	Almacena las distancias armónicas desde el tenor hasta el contralto y el soprano.	IM08

Id	Atributo	Función	Necesita
IA07	contraltoDistanceTable	Almacena las distancias armónicas desde el contralto hasta el soprano.	IM10
IA08	bassIntervalTable	Almacena la información de los intervalos armónicos desde el bajo hasta el tenor, el contralto y el soprano.	IM05
IA09	tenorIntervalTable	Almacena la información de los intervalos armónicos desde el tenor hasta el contralto y el soprano.	IM09
IA10	contraltoIntervalTable	Almacena la información de los intervalos armónicos desde el contralto hasta el soprano.	IM11
IA11	melodicTable	Almacena la información referente a los intervalos melódicos de cada par de notas.	IM06
IA12	chordsTable	Almacena la información referente a los acordes de la composición	IM07
IA13	fitness	Almacena el fitness del individuo en cuestión	FITNESS

tabla 11 - Individual: Atributos.

Con referencia a los métodos que implementa Individual, sólo se van a mostrar los que no pertenecen a la función de fitness, pues estos se explicaran en el siguiente apartado.

Id	Método	Parámetros	Tipo	Función	Necesita
IM00	__init__	Las codificaciones de las cuatro voces y la tonalidad.	Constructor	Crea un nuevo individuo a partir de las cuatro voces y la tonalidad.	Todos los IM e IA00, IA01
IM01	newRandom	La voz dada y la tonalidad.	Constructor	Crea un nuevo individuo aleatorio a partir de la voz dada y la tonalidad.	IM00, EM00
IM02	newID	La id del individuo, la tonalidad y la voz dada.	Constructor	Crea un nuevo individuo a partir de un id de individuo, la tonalidad y la voz dada.	IM00, IA00, IA01
IM03	tessituraTableMetohd	-	Objeto	Genera el atributo tessitura table.	VM04, VM05, VM06, VM07
IM04	distanceBassTableMetohd	-	Objeto	Genera el atributo bass distance table.	IA02, NM09, NM10
IM05	intervalBassTableMetohd	-	Objeto	Genera el atributo bass interval table.	IA02, NM07, NM08
IM06	melodicTableMethod	-	Objeto	Genera el atributo melodic table.	IA01, IA02, NM07, NM08, NM09
IM07	chordTableMethod	La tonalidad.	Objeto	Genera el atributo chords table.	IA01, IA02, IA05, NM00, TA01, TA03, TA04,
IM08	distanceTenorTableMetohd	-	Objeto	Genera el atributo .tenor distance table.	IA02, NM09, NM10
IM09	intervalTenorTableMetohd	-	Objeto	Genera el atributo tenor interval table.	IA02, NM07, NM08

Id	Método	Parámetros	Tipo	Función	Necesita
IM10	DistanceContraltoTable Metohd	-	Objeto	Genera el atributo contralto distance table.	IA02, NM09, NM10
IM11	intervalContraltoTableMetohd	-	Objeto	Genera el atributo contralto interval table.	IA02, NM07, NM08
IM12	individualFitness	-	Objeto	Genera el atributo fitness.	FM00

Tabla 12 - Invididual: Métodos Parte 1.

4.3.7. | Population.

En esta estructura se codifica la información referente al proceso evolutivo del algoritmo genético: la codificación de las poblaciones, inicialización de las mismas, operadores de selección, operadores de cruce, operadores de mutación, etc.

Los atributos correspondientes a Population son:

Id	Atributo	Función	Necesita
PA00	size	Almacena el tamaño de la población. Común para todas las poblaciones.	-
PA01	voiceGivenSet	Almacena la codificación de la voz dada.	-
PA02	bestIndividual	Almacena la información del mejor individuo encontrado en todo el proceso evolutivo.	-
PA03	individualSet	Almacena la información de individuos pertenecientes a la población.	-

Id	Atributo	Función	Necesita
PA04	bestIndividualPopulation	Almacena la información del mejor individuo de cada población.	PA03
PA05	worstIndividualPopulation	Almacena la información del peor individuo de cada población.	PA03

Tabla 13 - Population: Atributos.

Los métodos correspondientes a Population son:

Id	Método	Parámetros	Tipo	Función	Necesita
PM00	__init__	Lista con la información de los individuos pertenecientes a la población.	Constructor	Crea una nueva población a partir de la información de los individuos pertenecientes a esta	IA13
PM01	initialRandom	Tonalidad	Constructor	Genera una población inicial random.	PA00, PA01, PM00, IM01
PM02	changeSize Population	Tamaño de la población	Clase	Cambia el valor del tamaño de la población.	PA00
PM03	populationFitness	-	Objeto	Devuelve un set con el fitness de cada uno de los individuos de la población.	IA13, PA00, IM12
PM04	changeBestIndividual	Individuo	Clase	Se actualiza el mejor individuo del proceso evolutivo	PA02
PM05	Clash	Dos individuos	Clase	Devuelve el mejor individuo respecto a su fitness.	IA13

Id	Método	Parámetros	Tipo	Función	Necesita
PM06	tournament	Presión: número de individuos a tener en cada torneo.	Objeto	Realiza el proceso de selección de torneo en toda la población.	PA00, PA03, PM00, PM05
PM07	reproductionTwo Points	Dos individuos y la tonalidad.	Clase	Realiza el proceso de cruce simple sobre dos puntos entre dos individuos.	IA03, IM02, PA01
PM08	crossover	tonalidad	Objeto	Realiza el proceso de cruce en toda la población	PA00, PA03, PM00, PM07
PM09	selectiveSelection	-	Objeto	Aplica el proceso de selección elitista en la población	PA02, PA03, PA04, PA05, IA13
PM10	Mutation	Probabilidad de mutación y tonalidad	Objeto	Aplica el proceso de mutación sobre una probabilidad dada.	PA01, PA03, PM00, PM11, IA03, IM02
PM11	__probabilityMutation Positive	Probabilidad de mutación y tamaño del individuo.	Objeto	Devuelve la posición de los genes que deben mutar.	-

Tabla 14 - Population: Métodos.

4.4. | Fitness.

Se ha llegado a la función de fitness, punto esencial del proyecto, y para la que se han preparado todas las estructuras de datos anteriores (atributos y métodos); todo gira en torno a este concepto.

La propia hipótesis principal del proyecto se sustenta en poder otorgar un fitness, a cada individuo, que recoja: una evaluación de todas las reglas, aplicables en las Corales de Bach y presentadas en el Análisis del fitness, y las decisiones estéticas y correcciones sobre las reglas, extraídas de un maestro de la composición y docente de un conservatorio profesional de música.

Es por esto, que se realiza el fitness en dos fases:

- La primera consta de la codificación de la mayoría de las reglas analizadas, salvo ciertas reglas que se definirán una vez se hayan consultado en la entrevista con el compositor. Y se presentan una prueba de resultados.

- En la segunda fase, se realiza la comentada entrevista con el compositor y de los conocimientos extraídos se mejora el fitness inicial. Y se vuelven a presentar los resultados para ver en qué han afectado estos cambios.

4.4.1. | Fitness inicial.

Esta primera versión del fitness se va a basar en la tabla de las reglas analizadas en el apartado 3.1.2. Fitness, con ciertas excepciones.

En primer lugar, se va a realizar esta primera versión sin el requisito de la tabla de progresiones para estudiar cómo se comporta el algoritmo sin necesidad de establecerle unas progresiones obligatorias (F17).

Por la misma razón, no se han implementado la cadencia final obligatoria (F10).

La regla F18, referente al tritono, se ha decidido esperar a implementarla después de la entrevista, porque es necesario detallar adecuadamente el concepto.

Siguiendo esta preocupación, tampoco se ha implementado, en esta primera fase la estricción de octavas, quintas y unísonos directos (F05), ni las excepciones de las paralelas (F04).

Una vez sin progresiones los métodos generados en Individual para el cumplimiento de estas reglas son:

Id	Método	Parámetros	Tipo	Función	Necesita	Reglas
IM13	Croses	-	Objeto	Devuelve el número de cruces realizados entre voces.	IA08, IA09, IA10	F01
IM14	outTessitura	-	Objeto	Devuelve el número de notas fuera de la tessitura de su voz.	IA04	F19
IM15	notChords	-	Objeto	Devuelve el número de acordes que no existen	IA12	F02, F13
IM16	outDistance	-	Objeto	Devuelve el número de distancias armónicas más elevadas que la 8ª en las 3 voces superiores.	IA01, IA06, IA07	F03
IM17	fifthOctaves Parallels	-	Objeto	Devuelve el número de 5ª, 8º y unísonos paralelas.	IA01, IA05, IA06, IA07	F04
IM18	Augmented Seconds	-	Objeto	Devuelve el número de intervalos melódicos de segunda aumentada.	IA11	F06
IM19	outMelodic Intervals	-	Objeto	Devuelve el número de intervalos aumentados, de 7ª, de 9ª, o superiores.	IA11	F06, F07
IM20	voicesEqual Direction	-	Objeto	Devuelve el número de veces que se mueven todas las voces de un mismo acorde en una dirección similar.	IA11	F08
IM21	firstLastChord	-	Objeto	Devuelve el 1 o 2 si el primer y/o el último no son el primer grado en estado fundamental.	IA12	F09
IM22	Duplicate ProhibitedNote	-	Objeto	Devuelve el número de veces que se duplican notas con necesidad de resolver.	IA12	F11, F12
IM23	overLap	-	Objeto	Devuelve el número de sobrecruzamientos.	IA01, IA02, NM08	F16

Id	Método	Parámetros	Tipo	Función	Necesita	Reglas
IM24	resolutionNotes	-	Objeto	Devuelve el número de resoluciones de notas obligatorias fallidas.	IA02, IA08, IA11, IA12, NM08, NM09	F14, F15

Tabla 15 - Individual: Métodos Parte 2 (fitness V1)

Con esto se demuestra que todas las reglas están implementadas mediante los métodos de Individual.

Los atributos dentro de la estructura fitness son, se quita la columna de función ya que devuelven el parámetro total del método del que dependen, y se añade una nueva columna que incluye el peso por el que se multiplica el valor.

Id	Atributo	Peso	Necesita
FA00	nmOutTessitura	10	IM14
FA01	numCrosses	10	IM13
FA02	numNotChords	10	IM15
FA03	numOutDistance	10	IM16
FA04	numParallels	1	IM17
FA05	numAugmentedSeconds	1	IM18
FA06	numOutMelodicIntervals	1	IM19
FA07	numEqualDirection	1	IM20
FA08	numFirstLast	1000	IM21
FA09	numDuplicate ProhibitebeNotes	5	IM22
FA10	numOverLap	1	IM23
FA11	numNonResolution	10	IM24

Tabla 16 - Fitness: Atributos V1

Estos atributos están almacenados en el método:

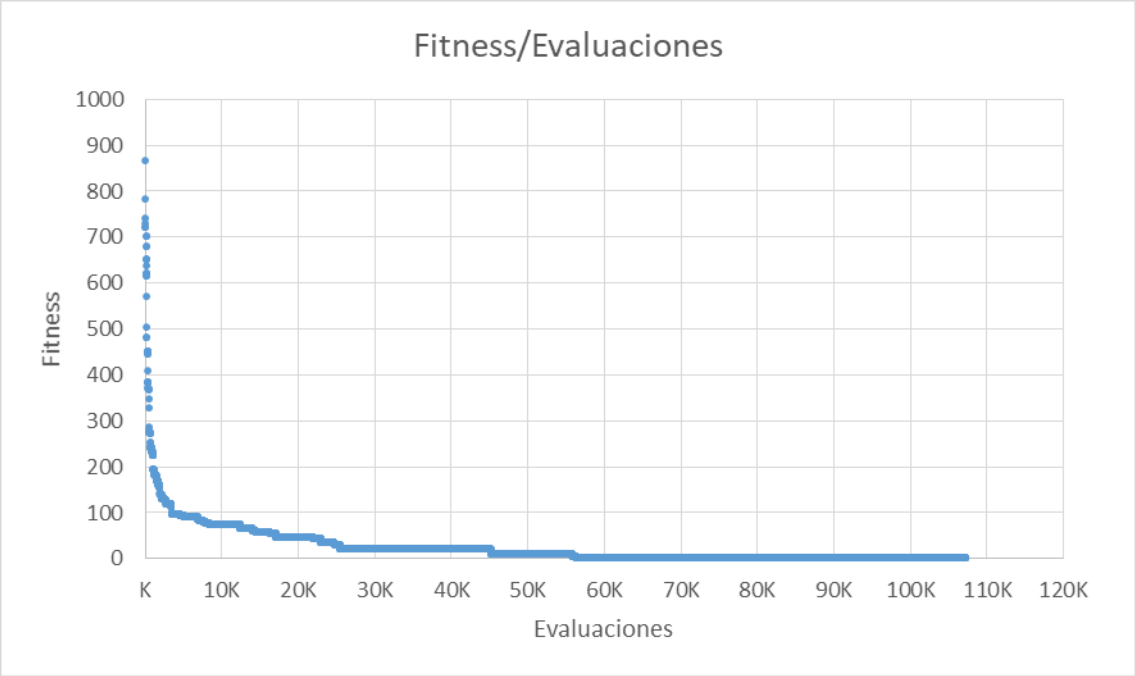
Id	Método	Parámetros	Tipo	Función	Necesita
FM00	fitness	Individuo	Clase	Devuelve resultado total del fitness del individuo.	Todos los FA

Tabla 17 - Fitness: Métodos.

Pruebas.

Una vez representados todos los métodos y atributos para dar funcionalidad a la primera versión del fitness se presentan los primeros resultados del algoritmo.

Las primeras pruebas son prometedoras, el algoritmo genético encuentra soluciones cuyos fitness rondan entre -5 y -1, en unas 6000 generaciones de media que logra generar en unos 20 minutos de ejecución.



Gráfica 1 - Fitness V1 Solo

Se puede observar la rápida mejora comentada anteriormente.

En principio, los resultados obtenidos con el algoritmo genético se pueden considerar muy buenos.

Pero, como era de esperar, respecto al analizar el resultado musical comete grandes errores en progresiones armónicas como comenzar y acabar en el primer grado con 7 o repetir grados en diferentes inversiones; y en el uso indebido de las reglas no implementadas hasta el momento, como la necesidad de acabar en una progresión armónica de V-I.

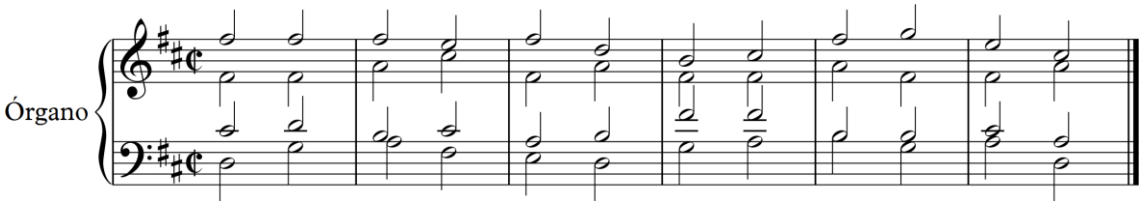


Figura 16 - Composición fitness V1

4.4.2. | Fitness V2: Retoque en las reglas.

Debido a los resultados obtenidos en las primeras pruebas del fitness, antes de incluir la codificación de loas progresiones de acordes permitidas, se plantea una mejora intermedia del fitness.

Esta mejora se va a centrar en retocar las reglas que ya se han tenido en cuenta para que las composiciones salgan más acordes con la forma musical buscada.

Frente a todas las pruebas realizadas se cree conveniente añadir las excepciones de las octavas, unísonos y quintas paralelas; de esta forma se busca ampliar las opciones resolutivas de la composición.

Por otro lado, se ha comprobado que, aunque las composiciones acaban y terminan en el primer grado, este debería estar en estado fundamental y ser una triada; por las pruebas realizadas, el algoritmo siempre tiende a las séptimas y novenas. Es por esto, que se ha retocado esta regla para obligar a que el primer y último acorde de la Coral sea una triada en estado fundamental sobre el primer grado de la tonalidad.

También, se ha incorporado la regla F10, en la que el penúltimo acorde debe ser en estado fundamental sobre la dominante de la tonalidad (el quinto grado). De esta forma, se obliga a generar una cadencia perfecta que da sensación de finalización y es de uso obligatorio en este tipo de composiciones.

Por último, como se entiende que las octavas, unísonos y quintas paralelas es de vital importancia que no se generen en las soluciones, ya que alteran en gran medida el estilo de la forma musical, se ha procedido a aumentar su peso dentro de la evaluación del fitness a 5.

Para implementar estos cambios, se ha generado un nuevo método en Individual:

Id	Método	Parámetros	Tipo	Función	Necesita	Reglas
IM25	Penultimate Chord	-	Objeto	Devuelve el número 1 si el penúltimo acorde no es fundamental sobre el V grado.	IA12	F10

Tabla 18 - Individual Métodos Parte 3 (fitness V2)

Con las últimas modificaciones la tabla de atributos del fitness sufre las siguientes modificaciones:

Id	Atributo	Peso	Necesita
FA00	nmOutTessitura	10	IM14
FA01	numCrosses	10	IM13
FA02	numNotChords	10	IM15

Id	Atributo	Peso	Necesita
FA03	numOutDistance	10	IM16
FA04	numParallels	5	IM17
FA05	numAugmentedSeconds	1	IM18
FA06	numOutMelodicIntervals	1	IM19
FA07	numEqualDirection	1	IM20
FA08	numFirstLast	1000	IM21
FA09	numDuplicate ProhibitebeNotes	5	IM22
FA10	numOverLap	1	IM23
FA11	numNonResolution	10	IM24
FA12	numPenultimateChord	1000	IM25

Tabla 19 - Fitness: Atributos V2

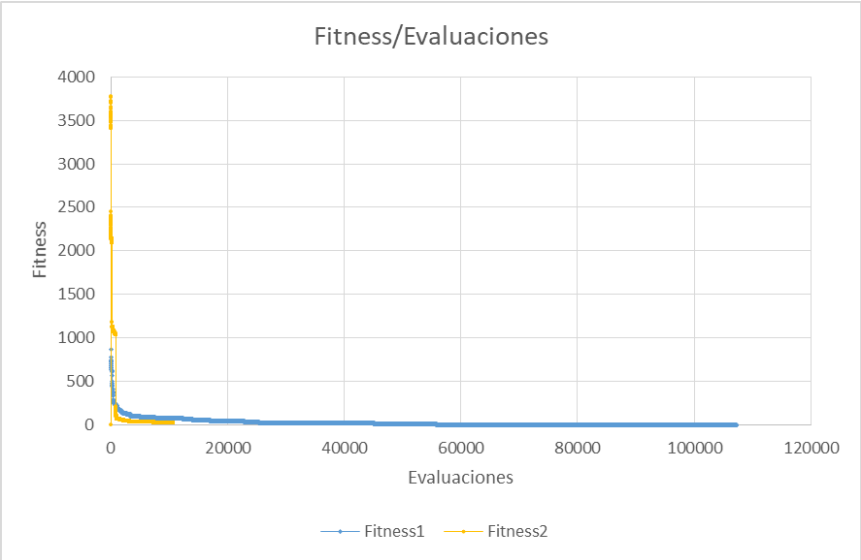
Pruebas.

Una vez preparada la nueva versión del fitness en el entorno de pruebas, se procede a generar varias pruebas y analizarlas.

Las pruebas generadas, suelen bajar de un fitness de -10 en torno a las 17.000 - 18.000 generaciones. Estos resultados eran de preveer ya que se ha reducido significativamente el ratio de soluciones posibles y al algoritmo de cuesta más llegar a ellas.

Uno de los resultados más significativos ha llegado a -1 en 56777 generaciones. Aunque este -1 se corresponde a un 0, ya que se encontró un error melódico en el bajo dado.

Gráfica 2 - Fitness V2 Sólo



Gráfica 3 - Fitness V2 Comparativa

Como se puede observar, la capacidad evolutiva del algoritmo se mantiene, y se ha generado una solución más apta ha aportado una composición mucho más en regla, frente al estilo de las Corales de Bach.

Cabe destacar, que se vuelve a extraer la necesidad de controlar las progresiones armónicas, ya que el principal fallo estilístico que se encuentra, reside en dichas progresiones generadas, como ya se había previsto, pero era necesario demostrar la hipótesis de dicha necesidad.

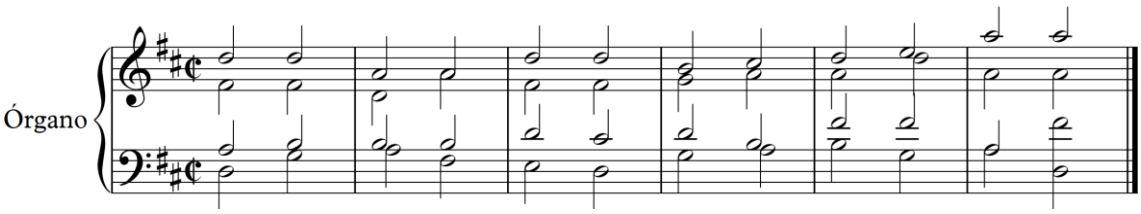


Figura 17- Composición Fitness V2

4.4.3. | Fitness V3: Progresiones Armónicas.

Una vez demostrado que el algoritmo no llega a crear verdaderas Corales de Bach si no se controlan las progresiones armónicas, se va a generar una nueva versión del fitness que incorpore la codificación de esta normativa.

Para ello se recuerdan las progresiones armónicas permitidas (el 6 hace referencia a la segunda inversión del acorde):

Grado	Posibles progresiones
I	TODOS
II	IV, V, VII
III	II6, IV, VI
IV	I, II, IV, I, VII
V	I, IV6, VI, VII
VI	II, IV, V, VII
VII	I, V, VI6

Tabla 20 - Progresión de acordes V2.

Sobre estas progresiones hay que incorporar la normativa en la que el primer grado puede estar en segunda inversión, si precede al quinto en la cadencia perfecta final, es decir, si es el antepenúltimo acorde.

Con este profundo recorte en el ratio de soluciones se prevé, un gran descenso en la eficacia del algoritmo, ya que será necesario un número mucho mayor de generaciones para alcanzar resultados viables.

Es por esto que se ha querido demostrar, antes de su incorporación al fitness, lo completamente imprescindibles que son estas normativas de progresiones para generar las composiciones deseadas.

Uno de las limitaciones que se esperaba correr dentro de este proyecto es, precisamente, ésta que se trata de exponer.

Al perseguir la perspectiva de un alumno de composición y abarcar el problema desde la completa aleatoriedad, las posibilidades son mucho más numerosas que si se acotara a una aleatoriedad sobre, únicamente, las notas de una tonalidad a tratar.

Pero, este es el objetivo del proyecto, analizar estas capacidades computacionales frente a este problema y llevar al límite la perspectiva; ya que una vez en el límite, en futuras investigaciones, es sencillo plantear recortes dentro del campo de búsqueda.

Para generar y controlar las progresiones se han implementado los siguientes métodos sobre Individual:

Id	Método	Parámetros	Tipo	Función	Necesita	Reglas
IM26	Chord Progressions	-	Objeto	Devuelve un diccionario (estructura de datos de Python) que contiene las posibles progresiones armónicas desde cada grado	-	F17
IM27	notChord Progressions	-	Objeto	Devuelve el número de acordes que no cumple las progresiones estipuladas.	IM26, IA12	F17

Tabla 21 - Individual: Métodos Parte 3 (fitness V3)

Los atributos añadidos al fitness son:

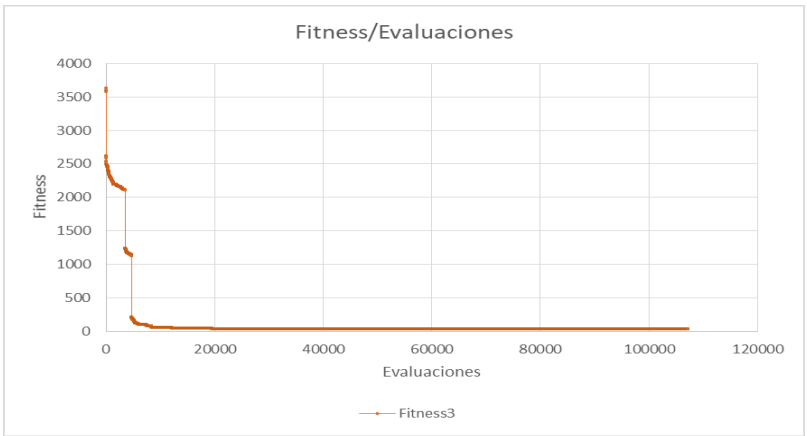
Id	Atributo	Peso	Necesita
FA13	numNotChordProgressions	5	IM27

Tabla 22 - Fitnes Métodos V3

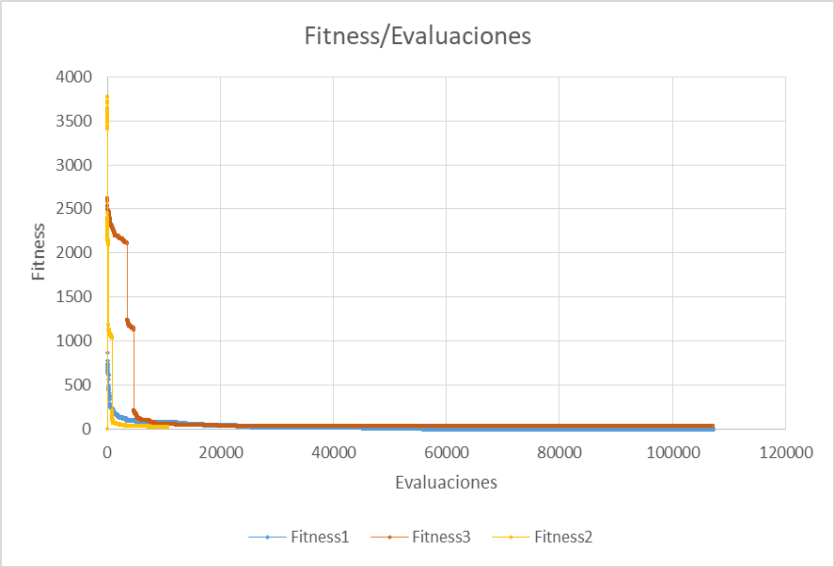
Pruebas.

Como ya se ha comentado, las pruebas presentan un número notablemente superior de generaciones necesarias para intentar asemejar los resultados de fitness respecto a las versiones anteriores.

De hecho, el mejor resultado encontrado es de -19 en 59795 generaciones y es por esto, que aunque la solución musical presenta progresiones más correctas, se encuentran demasiados errores para poder compararla con el resto.



Gráfica 4 - Fitness V3 Sólo



Gráfica 5 - Fitness V3 Comparativa.

Los resultados, respecto a las anteriores versiones del fitness, se pueden ver menos eficaces, por lo ya comentado.

4.4.4. | Entrevista con Álvaro Israel Gómez Alvarado.

El siguiente paso a dar en el estudio y diseño del fitness es reunirse con un maestro de la composición musical que retoque las reglas ya generadas e incorpore nuevos conceptos a codificar en el fitness. Para ello, se ha tenido la colaboración del gran maestro de la composición y profesor del conservatorio profesional de música de Leganés Álvaro Israel Gómez Alvarado.

Esta entrevista es de un valor vital para el proyecto, ya que permitirá representar el fitness de forma completamente fiel a la perspectiva de un músico de gran renombre.

Biografía.

Álvaro Israel Gómez Alvarado

Nacido en Madrid. Inicia sus estudios musicales en el conservatorio de Música "Rodolfo Halffter" de Móstoles en las disciplinas de violín con la profesora **Dolores Encina Guzmán** y solfeo con **Lucía Pérez Peláez**.

Posteriormente continúa sus estudios en el Conservatorio Profesional de Música "Teresa Berganza" con el profesor **Alejandro Saiz San Emeterio** en violín y con los profesores **Gabriel Fernández**, **Juan Carlos Panadero**, **Valentín Ruiz y Sebastián Mariné** en Armonía, Contrapunto, Composición y Acompañamiento; obteniendo el **Premio Fin de Carrera de Contrapunto y Fuga**, **Mención Honorífica Fin de Carrera de Armonía** y **Premio Fin de Carrera de Composición**.

Finaliza sus estudios de composición en el Real Conservatorio Superior de Música de Madrid con los profesores **Antón García Abril** y **Zulema de la Cruz.**

Como complemento a su formación ha realizado estudios de especialización en composición musical con **José Luis de Delás;** y ha asistidos a cursos impartidos por **Joan Guinjoan, Carles Guinovart, Román Alís, Tomás Marco, Cristóbal Halffter y Claudio Prieto,** entre otros.

Ha sido galardonado con el premio **"Jacinto e Inocencio Guerrero de Composición**" del **Real Conservatorio Superior de Música de Madrid** y finalista del concurso de composición de música coral **"Villa de Getafe"**.

Entre sus estrenos más destacados cabe reseñar la obra para piano titulada "**El Manantial**", que fue estrenada por el pianista Manuel Escalante durante el **homenaje realizado en la Universidad SEK de Segovia** a los compositores **Tomás Marco y Carmelo Bernaola** en el año 2003; la **ópera de cámara "Canción de Navidad"** basada en el Cuento de Navidad de Charles Dickens estrenada en el teatro **Mira** de Pozuelo en el año 2006.

Y como fruto de su colaboración con el compositor **Juan Olivera** con el que constituye "**Vengo toco y me voy"** ha realizado una gran multitud de músicas para imagen y teatro destacando títulos tales como **"Milagro en casa de los López"** dirigida por **Manuel Gancedo**, **"Bansia"** de **Carlos Pontini** dirigida por **Juanma Cifuentes, "Jean Louis y Lyly, el misterio de Madrid"** escrita y dirigida por **Pablo Vázquez, "Extremities" de William Mastrosimone** dirigida por **Pedro Casablanc , "De buena familia"** escrita y dirigida por **Natalia Hernández**, **"Pocohontas, el musical"** dirigido por María Pareja, **"Alelujah o el Jardín de las delicias"** escrita y dirigida por **Juan francisco Brihuega,"Los desterrados, hijos de Eva"** escrita por **Ana Fernández Valbuena** y dirigida por **Nacho Sevilla, "En la Colonia Penitenciaria"** dirigida por **Xavier Olza** y **Vicente Camacho, "Camas y Mesas"** escrita por **Emilio Williams López** y dirigida por **Isabel Pintor, "El volumen que te dan los rulos no te lo da un secador"** escrita por **Chema Trujillo** *y dirigida por* ***Mappy Laguna*** *y* ***Manuel Gancedo***, "**Te Quiero**" dirigida por **David Vélez**, entre otras.

Por otro lado, también desarrolla una intensa actividad como intérprete de música de cámara que le ha llevado a colaborar con diversas agrupaciones camerísticas con las que ha podido actuar en teatros y auditorios tales como el *Teatro Jovellanos* (Gijón), *Teatro Palacio Valdés* (Avilés), *Teatro Rojas* (Toledo), *Gayarre* (Pamplona), *Auditorio de Cuenca*, *Teatro Falla* (Cádiz),*Teatro- Auditorio de la Comunidad Gallega* (Santiago Compostela), *Teatro Caja de Ahorros de Cantabria*(Santander), *Bretón* de Logroño, etc.

Asimismo, ha trabajado en diversas obras de teatro como músico-actor de las que cabe destacar "**Una noche en el Canal"** con el director de teatro **Albert Boadella**, "**El Burgués Gentilhombre**" y "**Frankestein o el moderno Prometeo"** del director teatral **Gustavo Tambascio,** "**El volumen que te dan los rulos no te lo da un secador**" de los directores **Mappy Laguna y Manuel Gancedo**, "**Carro de Rueda**" dirigida por **Emilio Lorenzo**, entre otras.

Por último, desarrolla una intensa labor docente, habiendo sido profesor en diversos conservatorios e instituciones; siendo en la actualidad profesor de lenguaje musical, armonía, análisis y fundamentos de la composición en el conservatorio profesional de música **"Manuel Rodríguez Sales"** de Leganés.

Entrevista.

Una vez se ha descrito una breve biografía del compositor Álvaro Israel Gómez Alvarado, se procede a explicar en qué ha consistido la entrevista y los conocimientos extraídos de la misma.

En primer lugar, se ha hablado con el compositor, para explicarle en qué consiste el proyecto, los objetivos y la perspectiva que le busca dar a la investigación. Además, se le platea el gran interés en su colaboración, para poder extraer una perspectiva compositiva aún más fiel y que el fitness se convirtiera en una codificación de su visión frente a las Corales de Bach.

Él desde el primer momento se ha mostrado completamente dispuesto a colaborar en todo lo necesario con el proyecto, es más, de cara a la presentación se ha planteado poder presentar una Coral Compuesta por él, enfrentada a una solución del algoritmo; que ha ser posible, él también habrá corregido.

En cuanto a la entrevista en sí, se ha planteado de forma técnica, con la tabla de reglas implementadas, se han ido comentado todas y cada una de ellas; seguidamente se le han hecho ciertas preguntas para extraer todo el conocimiento posible de esta oportunidad de colaboración entre ciencia y arte.

Respecto a las reglas implementadas en la última versión del fitness, comenta ciertos detalles a tener en cuenta.

- Los **cruces de voces**, son algo que, en ocasiones los compositores pueden llegar a permitir, es por ello que necesitan tener un peso mucho menor en la evaluación.
- El único acorde permitido en **segunda inversión es el I** cuando acompaña a la cadencia perfecta final.
- Las **séptimas de sensible y disminuidas** (VII grado con 7ª) no se utilizan en las este tipo de formas musicales ya que aportan demasiado color a la pieza y en el Barroco esto no estaba bien visto.
- Por esta misma razón tampoco se deben permitir los **acordes de novena.**
- Respecto a las dudas que se tenían con el **concepto de tritono**, nos explica que es el intervalo de 4ª aumentada, mal visto en la época, y que ya lo estamos contemplando en las reglas, ya que no se permiten los intervalos aumentados.
- Explica que las **octavas, unísonos y quintas directas** prohibidas, consisten en la realización del mismo movimiento entre las voces externas y que este movimiento de lugar a un intervalo armónico entre dichas voces de 8ª, 5ª o unísono. Esto tiene la excepción de que el movimiento de la voz soprano se mediante grado conjunto, es decir, un intervalo de segunda.

- También ve necesario, analizando las soluciones generadas por el algoritmo, **restringir el número de intervalos melódicos de unísonos** a no más de dos seguidos.

Una vez vistas las reglas, se le han realizado las siguientes preguntas.

- **¿Se podrían usar otros tipos de acordes como el napolitano o las sextas francesas?** La respuesta es negativa y explica que se volvería al problema de dar demasiado color a la composición.

- **¿Se permite alguna progresión que no figure en la tabla analizada en los libros?** En principio, es clave respetar estas restricciones armónicas, ya que se trata de una forma musical muy estructurada.

- **De cara a futuras mejoras del algoritmo, ¿Qué conceptos musicales incorporarías?** En primer lugar comenta, que el siguiente paso es introducir las modulaciones, ya que en la forma música de las Corales de Bach, están muy limitadas a las tonalidades vecinas y se producen en notas más largas que el resto; por eso, cree que podría llegar a ser relativamente sencillo sintetizar esa idea de forma lógica y estructurada.

  También, comenta que incorporaría otros tipos de cadencias, pero que para realizarlas hay que conocer los puntos donde la música descansa, como en los calderones (símbolo que alarga la duración de una nota).

Finalmente, la entrevista finaliza quedando en lo propuesto anteriormente, proporcionará, para la presentación, una Coral compuesta por él y analizará las soluciones que ha sido capaz de generar el algoritmo frente a ese bajo o tiple.

4.4.5. | Fitness Final: Aplicando lo extraído de la entrevista.

Una vez se ha realizado la entrevista con el compositor Álvaro Israel Gómez Alvarado, se han añadido los conocimientos extraídos en forma de nuevas reglas a codificar:

Id	Regla	Excepciones	Id: Tipo de Información
F05	Se permiten 5ª, 8ª o unísonos directas	En voces externas y la soprano no viene de un intervalo de 2ª.	07, 08
F20	Se prohíben los acordes de séptima se sensible y disminuida.	-	05
F21	Se prohíben los acordes de novena.	-	05

Id	Regla	Excepciones	Id: Tipo de Información
F22	Se prohíben la segunda inversión.	Primer grado que precede a cadencia perfecta final.	05
F23	Se prohíbe repetir el intervalo melódico de unísono más de dos veces.	-	06

Tabla 23 - Fitness Final: Reglas.

Para poder implementar estas últimas reglas se crean los siguientes métodos sobre Individual:

Id	Método	Parámetros	Tipo	Función	Necesita	Reglas
IM28	fifthOctaves Direct	-	Objeto	Devuelve el número de unísonos, octavas y quintas directas.	IA01, IA05, IA11	F05
IM29	notSeventh7	-	Objeto	Devuelve el número de acordes VII7	IA12	F20
IM30	notNinth	-	Objeto	Devuelve el número de acordes de novena.	IA12	F21
IM31	Not64	-	Objeto	Devuelve el número de acordes en segunda inversión. Teniendo en cuenta la excepción del I grado que precede a la cadencia perfecta final	IA12	F22
IM32	noMoreThan twoUnison	-	Objeto	Devuelve el número más tres unísonos melódicos seguidos	IA11	F23

Tabla 24 - Individual: Métodos Parte 4 (fitness final)

Los atributos añadidos y los cambios realizados en los pesos sobre fitness son:

Id	Atributo	Peso	Necesita
FA00	nmOutTessitura	10	IM14
FA01	numCrosses	10	IM13
FA02	numNotChords	10	IM15
FA03	numOutDistance	10	IM16
FA04	numParallels	5	IM17
FA05	numAugmentedSeconds	1	IM18
FA06	numOutMelodicIntervals	1	IM19
FA07	numEqualDirection	1	IM20
FA08	numFirstLast	1000	IM21
FA09	numDuplicate ProhibitebeNotes	5	IM22
FA10	numOverLap	1	IM23
FA11	numNonResolution	10	IM24
FA12	numPenultimateChord	1000	IM25
FA13	numNotChordProgressions	5	IM27
FA14	numFOdirect	1	IM28
FA15	numNotSeventh7	5	IM29
FA16	numNotNinth	10	IM30
FA17	numNot64	10	IM31
FA18	numMoreTwoUni	1	IM32

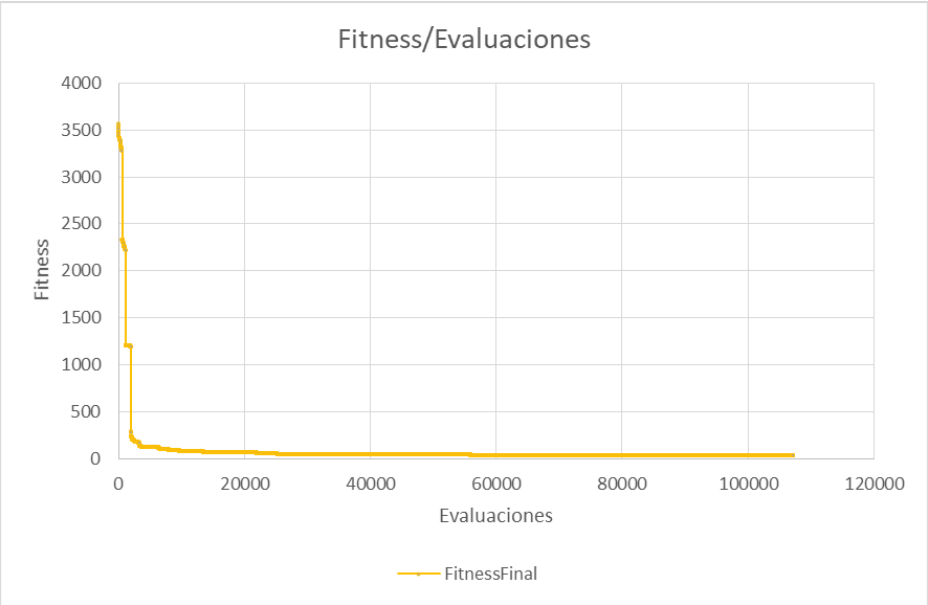
Tabla 25 - Fitness Final: Atributos.

Pruebas.

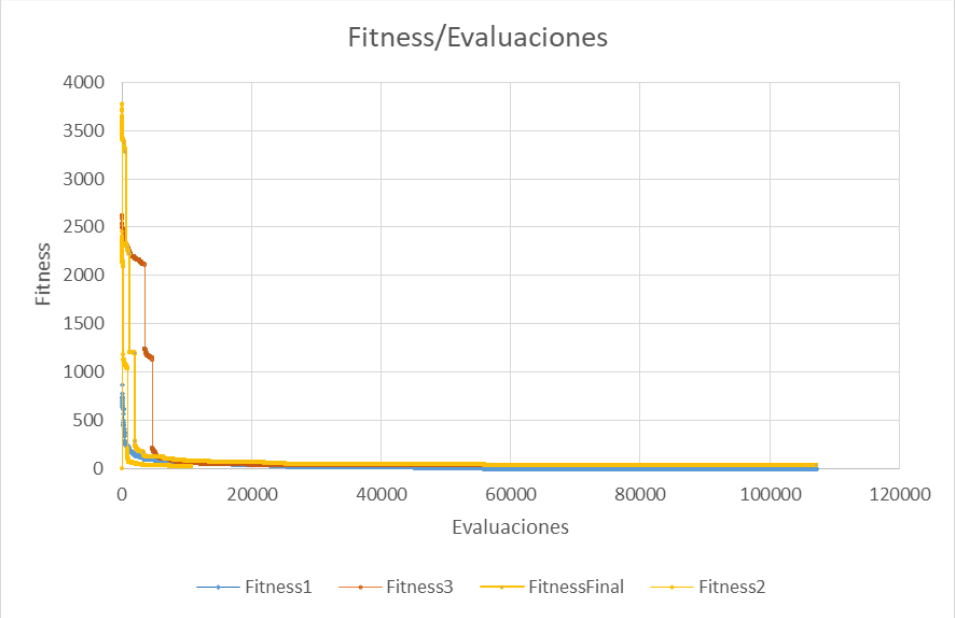
Se analiza un mejora respecto a la versión anterior del fitness, en la eficiencia del algoritmo, esto es debido a la implentación de ciertas reglas que elevan las posibles soluciones, a su vez, otras reglas más restrictivas han aportado una línea de búsqueda de soluciones más coherente para el algoritmo.

La mejor prueba a obtenida ha sido bajo un fitness de -12 en unas 100000 generaciones.

Debido a un fitness mínimo de -12, no se cree instructivo pararse a analizar la parte musical, más allá de que se vislumbra un avance claro hacia las soluciones compositivas del puro estilo de las Corales de Bach



Gráfica 6 - Fitness Final Sólo.



Gráfica 7 - Fitness Final Comparativa.

4.5. | Mejoras sobre el Algoritmo Genético.

Una vez ya se ha cerrado el diseño del fitness, se ha comprobado que una composición, con este elevado número de restricciones; y desde una perspectiva de realizar una búsqueda desde cero, y sobre un espacio que englobe todas las posibles notas dentro de 4 octavas; supone una gran limitación para la eficiencia del algoritmo genético.

Es por esto que se va a buscar las posibles mejoras, empezando por el estudio de las diferentes combinaciones de sus parámetros y operadores, y terminando con nuevos posibles diseños para el proceso evolutivo.

Para cada una de las decisiones, se realizan 10 pruebas sobre cada posibilidad y se muestra, en una tabla, el mejor fitness encontrado entre todas las pruebas, el peor y el término medio de los 10 fitness; por otro lado, se muestra el tiempo de ejecución. Todas las pruebas se realizan sobre 4000 generaciones, que, por todas las pruebas realizadas, ya se puede ir deduciendo la eficacia del algoritmo.

4.5.1. | Tamaño de población.

Se van a analizar tamaños muy pequeños (4, 5, 8 y 10) y tamaños elevados (20).

Tamaño	Mejor Fitness	Fitness Medio	Peor FItness	Tiempo Aproximado (seg)
4	-14	-890,3	-2058	360
5	-32	-297,5	-1049	549
8	-54	-397,3	-1048	690
10	-40	-66,3	-83	880
20	-56	-72,6	-84	1800

Tabla 26- Pruebas Tamaño de Población

Se puede comprobar que en poblaciones grandes (20) el coste de tiempo es bastante insostenible y los resultados no son buenos respecto al resto de opciones.

Por otro lado, se observa que, aunque el resto de opciones están muy igualadas, la mejor decisión se puede comprobar en la gráfica y en la tabla que reside bajo el tamaño de población de valor 10.

4.5.2. | Redundancias en la codificación.

Para resolver las redundancias analizadas en la codificación, se puede optar por tres soluciones: nota aleatoria, do natural, primera nota de la tonalidad.

Solución	Mejor Fitness	Fitness Medio	Peor FItness
Nota aleatoria	-124	-204,6	-2048
Do natural	-40	-66,3	-83
1ª Nota	-96	-540	-1024

Tabla 27 - Pruebas redundancia codificación.

Como se había previsto, al ser do natural la nota que más se repite y más común en todas las tonalidades, el mejor método para tratar la redundancia de la codificación es sustituyéndola por el Do natura.

4.5.3. | Tipo de torneo.

En lo que respecta a la selección, se ha diseñado que sea por torneo, ahora es necesario analizar si es mejor realizar el torneo entre 2, 3 o 4 individuos.

Nº Individuos	Mejor Fitness	Fitness Medio	Peor FItness
2	-36	-151.8	-256
3	-40	-66,3	-83
4	-200	695,4	-1098

Tabla 28 - Pruebas Tipo Torneo.

Por lo que se ha podido analizar en las pruebas y se observa en los resultados, el torneo entre dos y tres individuos aporta buenos resultados, pero se cree conveniente quedarse con 3 individuos para que exista mayor competencia en la selección.

4.5.4. | Tipo de cruce simple.

El cruce que se lleva realizando en todas las pruebas es sobre dos puntos, pero es necesario analizar si es mejor o peor que sobre un único punto.

Puntos	Mejor Fitness	Fitness Medio	Peor FItness
1	-98	-153,6	-204
2	-40	-66,3	-83

Tabla 29 - Pruebas Tipo Cruce.

Los resultados obtenidos son bastante similares, pero, por el análisis hecho anteriormente, en el que se explica la relación estructural de tres partes de las corales con el cruce por dos puntos, se cree conveniente mantener el cruce sobre dos puntos.

4.5.5. | Probabilidad de mutación.

La mutación se realiza bit a bit sobre una prioridad de 0.05. Es necesario analizar si es mejor una probabilidad de 0.1 o, por el contrario, una probabilidad que se reduzca según pasa el tiempo.

Probabilidad	Mejor Fitness	Fitness Medio	Peor FItness
0.05	-40	-66,3	-83
0.1	-72	-544,4	-1035
Variable	-84	-603,3	-2023

Tabla 30 - Pruebas Mutación.

Como se puede analizar, los mejores resultados se obtienen bajo una probabilidad de mutación de 0.05.

4.5.6. | Mejora con Nichos.

El problema que se presenta en este proyecto, no tiene una única solución, si no que diferentes composiciones sobre la voz dada pueden obtener un fitness de 0, es decir, ser completamente válidas.

Es por esto, que se plantea una mejora del proceso evolutivo mediante el concepto de Nichos.

Los llamados Nichos permiten que el proceso de evolución no se centre en una única línea de mejora, si no en varias.

Para conseguir esto, se basa en el porcentaje de diferencia entre unos individuos y otros; este porcentaje se multiplica por el fitness.

Es decir el fitness de un individuo es: **fitness * nº de individuos parecidos en un 98%.** Si se analiza que funciona, se puede analizar con qué porcentaje de similitud, el algoritmo, funciona mejor.

Esta mejora se deja planteada para llevarla a cabo en el futuro.

4.5.7. | Mejora con Islas.

Como futura posible mejora, se plantea un diseño del proceso evolutivo mediante islas. Este nuevo diseño consistiría en evolucionar 3 poblaciones en paralelo, dos de ellas de poblaciones de 8 individuos que evolucionan de forma independiente una de la otra; y la tercera se usaría para enfrentar a las otras dos poblaciones, generándose a partir de los 4 mejores individuos de cada una de las otras poblaciones.

De forma gráfica consistiría en:

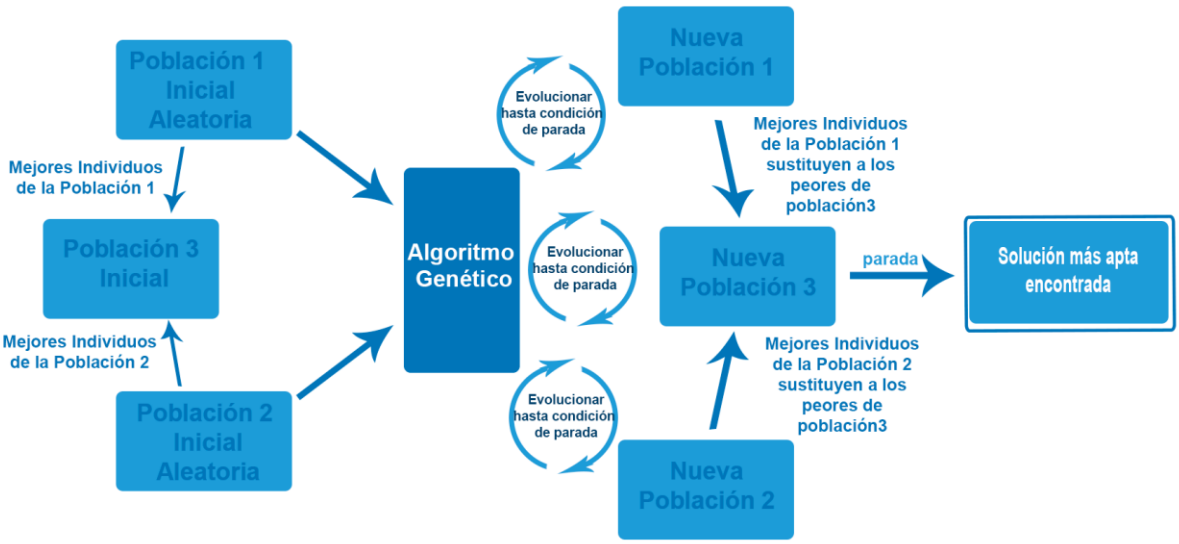


Figura 18- Algoritmo Genético Con Islas.

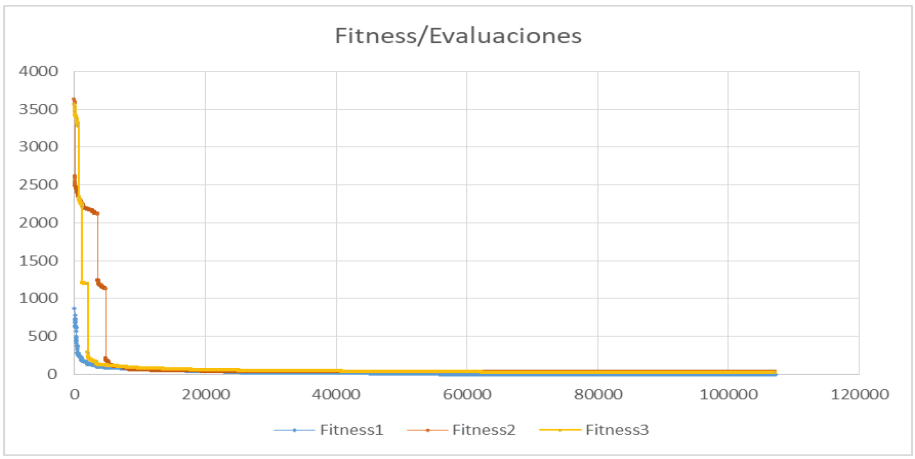
Esta mejora se deja planteada para llevarla a cabo en el futuro.

05. | PRUEBAS FINALES.

Como se ha ido analizando el diseño final del algoritmo consiste en poblaciones de 10 individuos, con selección por torneo de 3, con cruces simples por dos puntos y mutaciones bajo una probabilidad de 0.005 de ocurrencia.

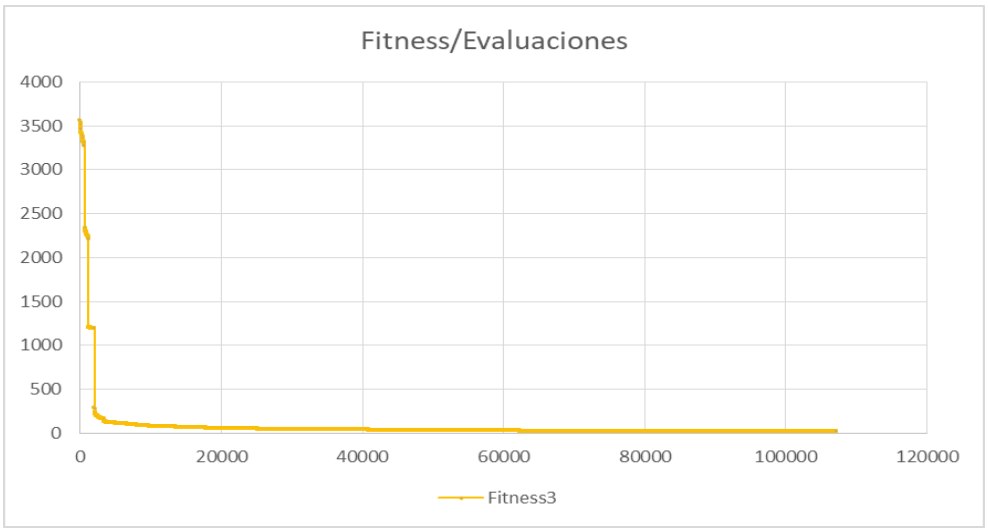
Todo esto aporta unos resultados válidos pero, en la mayoría de las pruebas, ineficientes.

Se comparan todos los avances obtenidos en la siguiente gráfica:



Gráfica 8 - Prueba Final Comparativa.

Siendo la gráfica de una de las pruebas sobre el diseño final:



Gráfica 9 - Prueba Final Sólo.

Se puede comprobar una mejora en estas últimas pruebas, pero la mejora de la eficiencia se planteara en la presentación de proyecto, junto con las mejoras del concepto de nichos y/o islas.

En lo referente a la parte de las pruebas del estilo musical:

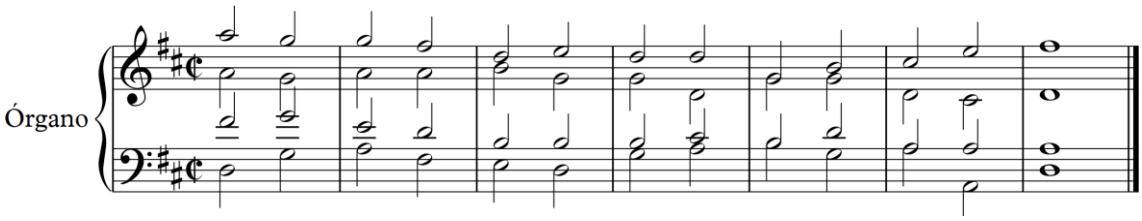


Figura 19- COmpasición Prueba Final.

Se puede analizar una increible mejora respecto a las primeras versiones del algoritmo. A pesar de tener un fitness de -12 capta notablemente mejor el estilo de las corales de Bach que los aquellas primeros resultados que conseguían fitness inferiores a -5.

5.1. | Entrevista con Claudio Tupinambá.

Para concluir el proyecto, se le propuso, al maestro Claudio Zagari Tupinambá, realizar una entrevista en la que se le presentara y explicara el proyecto y se le hicieran ciertas preguntas sobre que le parecía desde la perspectiva de un músico o si le encuentra utilidad a la investigación dentro de su profesión como músico y docente.

Antes de pasar a la entrevista en sí, se presenta una breve biografía del entrevistado.

Biografía.

Claudio Zagari Tupinambá.

Nació en 1970 en la ciudad de Rio de Janeiro, Brasil. Comenzó a formarse como músico a muy temprana edad (7 años), obteniendo entre los años 1989 y 1992 el **Título Superior de Guitarra** en la Escuela de Música perteneciente a la Universidad Federal de Río de Janeiro, bajo la docencia de **Leo Soares**.

Entre 1992 y 1993, obtuvo los títulos de postgrado del **Real Conservatorio Superior de Música de Madrid** con **Jorge Ariza** y en la Hochschule für Musik (**Mozarteum**) de Salzburgo, Austria durante los años 1993 y 1997, bajo la docencia de uno de los grandes nombres de la guitarra, **Eliot Fisk.**

Sus estudios de composición fueron bajo los nombre de **H.J.Koellreuter, Mario Ficarelli y Leo Brouwer.**

Como intérprete, guitarrista, da numerosos conciertos por Europa, América Latina, Marruecos, EEUU, etc. Solista de grandes orquestas con maestros de la talla de **Henrique Morelembaum, Ligia Amadio o Ciro Tabet.** Uno de los conciertos que es necesario destacar es el dado en el 75º Aniversario de la Biblioteca Musical de Madrid en 1994, tocando una de las guitarras que pertenecieron a **Andrés Segovia**.

Su obra ha sido interpretada por todas partes del mundo y grabada en diferentes discos. Las obras más destacables son **Around the FIre, Acordes, Toccatta** o las recientes **Darwin is Wrong** o **Viral.**

Ha publicado un CD como solista llamado **Mosaico** que busca hacer un giño a la música brasileña contemporánea para guitarra. Por otro lado, ha colaborado en otros CD junto con maestros tales como **Egberto Gismonti, la London Simphony Orchestra, etc.**

En la actualidad es profesor de guitarra en el **conservatorio profesional de música Pablo Casals de Leganés.**

Entrevista.

Respecto a la entrevista, se ha realizado en dos días.

En el primero se le presenta el proyecto y se le explica cómo funciona y la perspectiva que se la ha intentado dar, buscando sintetizar las acciones compositivas de un músico.

Esto da pie a un debate sobre la tecnología y su uso dentro de las diferentes ramas del arte. Estando ambos de acuerdo en la necesidad de colaboración entre ciencia y arte para encontrar puntos en común y poder ayudarse mutuamente.

El segundo día, se vuelve a recordar lo referente a los objetivos, perspectiva y funcionamiento del algoritmo diseñado, y se le realizan las siguientes preguntas.

¿Usarías un programa que te ayudase a componer?

Explica que debido a su proceso de composición, es complicado introducir un software que le llegue a aportar lo que el necesita; ya que cuando compone disfruta sumergiéndose en sí mismo y explorando cual intruso dentro de su propia creatividad.

Pero, añade que si surgiera la necesidad, no tendría ningún reparo en hacer uso de cualquier tipo de software.

¿Crees que la música se puede estructurar de forma matemática?

Él explica que la música se basa en la matemática, eso es algo que se ha de tener claro; una partitura es pura matemática. Pero, el entiende que a la hora de componer existe un componente espiritual, que no llegamos a entender.

Ese componente se refleja en todo tipo de arte, con él se logran evocar vivencias personales que, en cierto modo, pueden llegar a ser empíricas, pero de una complejidad muy elevada para los conocimientos actuales.

¿Dónde crees que reside este límite, del que hablas, en el que no se llega a poder objetivar el arte?

Comienza su respuesta explicando que si tú miras un Retrato de Rembrandt, observas multitud de técnicas objetivas: la luz, la composición de colores, las dimensiones, etc. Pero, la sensación que genera el mirar a los ojos de ese hombre retratado, es totalmente indefinible e irreplicable.

La percepción humana es difícilmente medible, actualmente y expone la creencia de que se está lejos de lograrlo.

Seguidamente, explica que el lenguaje funciona porque hay dos personas concuerdan entre sí, por ejemplo, el humor de España, cuando lo usa en Brasil, no se entiende.

El arte del siglo XX genera unas nuevas matemáticas al servicio de sus creaciones, por ejemplo, el dodecafonismo de Semper. Estas composiciones no son entendibles si no entiendes la matemática que se ha creado para ellas.

Una vez creado el dodecafonismo es perfectamente analizable y sintetizable, pero lo que no lo es, fue el proceso de creación que llevo a Semper a crear esa forma musical.

Por último, remarca que es necesario diferenciar dos lados del arte igualmente bellos y genuinos. Por un lado, el arte como el del David de Miguel Ángel, es perfecto, una perfección técnica. En cambio los retratos de Rembrandt, también son perfectos, pero su perfección no reside en la técnica, si no en algo que no alcanzamos a comprender.

¿Qué aportes puede tener, a la comunidad musical, esta investigación y las demás que se hacen en este campo?

Comienza diciendo que a él ya le ha aportado; ver la perspectiva de la inteligencia humana a la luz de la inteligencia artificial, te da conciencia de tu capacidad creativa y los procesos mentales que se realizan en la composición. Es como generar un espejo de uno mismo.

En este algoritmo has logrado de forma muy fiel a los principios de un compositor, la forma de componer las Corales de Bach.

En la actualidad, explica que se vive en una modernidad líquida, en la que las nuevas generaciones buscan el experimentar. Aprenden de otra forma, no quieren memorizar, quieren explorar lo máximo posible para aprender lo que realmente necesitan.

Él mismo, explica, está utilizando en sus clases de solfeo, dispositivos móviles con programas que apoyan las clases y ha notado un notable incremento en el interés de los alumnos, sus notas y el dinamismo de las clases.

Haciendo un símil con los ordenadores, explica que los nuevos alumnos buscan ser memorias RAM, no discos duros.

Es por esto que cree que investigaciones de este tipo pueden dar lugar a herramientas que apoyen este carácter de experimentación de los alumnos, y, a su vez, a los profesores, por ejemplo, proporcionándoles muchas más variables en la solución a un ejercicio que en el tiempo de la clase sólo le daría tiempo a analizar de una única forma, como mucho de dos.

La entrevista, fue un diálogo intenso, como se ha podido comprobar, pero se extrae fácilmente, la idea clara de la perspectiva de un músico frente a estas investigaciones y la necesidad de colaboración entre ciencia y arte, para conseguir llegar a los recónditos escondites de aquella percepción humana que aún no hemos llegado a entender.

06. | GESTIÓN DEL PROYECTO.

La gestión del proyecto engloba todo los campos referentes a la administración de dicho trabajo.

El **entorno socio-económico** donde se define el campo de aplicación de la investigación y su impacto en el mismo.

El **marco regulador** donde se explica la licencia que se le otorga a este proyecto.

La **planificación**, se detallan las fases que se han llevado acabo para la realización del proyecto, desde que se plantea la propuesta hasta que se presenta. Se hace uso de un diagrama de Gantt para mostrar el proceso de desarrollo, donde si visualizan las fases colocadas en una línea temporal

Los **presupuestos,** se desgranan todos los tipos de gastos que conlleva el proyecto (software, hardware, viajes y dietas, gastos indirectos y recursos humanos); para, finalmente, concluir con el presupuesto final del proyecto que tiene en cuenta todos los gastos mencionados anteriormente.

6.1. | Entorno Socio-Económico.

Este proyecto está vinculado al apoyo de la educación musical de los conservatorios y se busca realizar un aporte dentro del campo de investigación de la generación de música evolutiva.

6.1.1. | Educación musical.

Con la realización de esta investigación se busca llegar a entender el proceso compositivo de un músico dentro de la forma musical de Coral de Bach.

Con esto se pretende poder dar un apoyo a los profesionales del campo de la música, principalmente docentes, para que puedan aplicar dichos conocimientos a la hora de educar nuevos músicos en el campo de la composición.

Por otro lado, si se le añade una interfaz cómodo para los usuarios docentes y alumnos de composición, les permitiría poder generar soluciones muy variadas frente a un mismo problema; que como ya se comentó en la entrevista con el maestro Claudio Tupinambá, esto daría la posibilidad al profesor de poder explicar sobre una infinidad de variedad de soluciones y a los alumnos poder nutrir su curiosidad autodidacta y “trastear” con las soluciones.

A esto se le puede añadir la posibilidad de generar una aplicación que corrija soluciones de corales de Bach con el fitness ya creado. De esta forma permitirá experimentar a los alumnos de forma libre.

6.1.2. | Investigación.

En el entorno de la investigación se busca, con este proyecto, abrir un camino hacia una nueva línea de investigación, donde se busque entender los procesos mentales de un compositor desde la pura objetividad.

Para ello, este proyecto da el primer paso, estudiando y sacando conclusiones de la primera forma musical que estudia un compositor en su proceso de aprendizaje.

6.1.3. | Entorno económico.

Como ya se ha explicado este proyecto no se plantea de cara a poder introducirlo en el entorno económico. De hecho, como se explica en el apartado de marco regulador, que se presenta a continuación, se le otorga una licencia que impide el uso para fines lucrativos del proyecto.

Aunque, de forma indirecta, los conocimientos aportados pueden llevar a herramientas y/o aplicaciones como las ya mencionadas que en un futuro podrían introducirse en este entorno.

6.2. | Marco Regulador.

Las licencias **Creative Commons**. [23] son modelos de licencias libres que dan la posibilidad de compartir los proyectos y otro tipo de obras en internet de forma libre, pero con una limitación de los posibles usos que el resto de personas pueden hacer de la obra o proyecto.

Las diferentes restricciones de uso que se pueden utilizar son:

Reconocimiento: Se permite cualquier uso de la obra de forma pública, siempre y cuando se reconozca la autoría original del mismo.

Sin obras derivadas: Se restringe el uso de la obra para generar otras obras derivadas de ésta.

No comercial: No se permite ningún uso de la obra con fines lucrativos.

Compartir Igual: Se permite cualquier uso de la obra, siempre y cuando, este sujeto bajo la misma licencia.

Con estas cuatro restricciones se pueden generar seis licencias diferentes:

- **Reconocimiento:** Permite cualquier uso de la obra siempre y cuando se reconozca su autoría original.
- **Reconocimiento-SinObrasDerivadas:** Permite cualquier uso de la obra, excepto generar otras obras derivadas, y con la condición de reconocer su autoría original.

- **Reconocimiento-SinObrasDerivadas-NoComercial:**Permite cualquier uso de la obra no lucrativo, excepto generar otras obras derivadas, y con la condición de reconocer su autoría original.
- **Reconocimiento-NoComercial:** Permite cualquier uso de la obra no lucrativo, con la condición de reconocer su autoría original.
- **Reconocimiento-CompartirIgual:** Permite cualquier uso de la obra siempre y cuando se reconozca su autoría original y se utilice la misma licencia que la obra original.
- **Reconocimiento-NoComercial-CompartirIgual:** Permite cualquier uso de la obra no lucrativo, con la condición de reconocer su autoría original y se utilice la misma licencia que la obra original.

Estas licencias otorgan el derecho del autor a acudir a los tribunales en caso de su incumplimiento; pero, en ningún caso, aumentan o disminuyen los derechos otorgados por la ley al autor.

La licencia otorgada al proyecto es del tipo "**Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional**" de Creative Commons.

Más concretamente restringe el uso comercial del material; en el caso de ser utilizado en otros proyectos se debe referenciar su autoría indicando los cambios realizados y los proyectos que deriven de éste deben utilizar la misma licencia.

Por otro lado, todas las librerías propuestas están sujetas bajo licencias **GNU GPL** (GNU General Public License).

Este tipo de licencias de derechos de autor otorga la garantía de que la obra es de libre uso. El objetivo de estas licencias es promover el software libre y, a su vez, proteger la autoría general del mismo.

Por último, tanto Claudio Tupinambá como Álvaro Israel Gómez Alvarado han dado su permiso para publicar dentro de este documento tanto su imagen, como sus nombres y biografías, además de las entrevistas realizadas a cada uno de ellos.

6.3. | Planificación.

La planificación de este proyecto se puede agrupar en 9 fases:

- **Planteamiento del Proyecto:** En esta fase, el autor Carlos Bragado Sánchez, se reúne con su tutor Yago Sáez para plantear el problema y planificar los principales pasos a seguir y los objetivos a alcanzar.

- **Análisis del Estado del Arte:** En esta fase, se recopila toda la información posible acerca del campo de estudio y se analiza, primero buscando una visión general y posteriormente haciendo un trabajo de análisis más profundo de los proyectos para buscar ideas de algoritmos y diseños de posibles soluciones.

- **Desarrollo del fitness general:** Una de las fases principales, se realiza casi desde el principio del proyecto ya que es necesario saber con la mayor brevedad posible si es posible codificar todas las reglas de las Corales de Bach.

- **Desarrollo del Algoritmo Genético:** Se desarrolla una primera versión del algoritmo genético para probar que el fitness es adecuado y los resultados de las primeras pruebas indican que la investigación va por buen camino.

- **Mejora del fitness:** Se realiza una entrevista, con el maestro compositor Álvaro Israel Gómez Alvarado, para mejorar las reglas del fitness, corregir posibles errores conceptuales en ellas y agregar nuevas reglas subjetivas de forma codificada al fitness.

- **Mejoras en el Algoritmo Genético:** Una vez se ha dado por cerrado el fitness, se busca mejoras para el algoritmo genético. Esta fase se plantea su continuación posterior a la entrega de este documento, hasta la fecha en la que se presenta el proyecto.

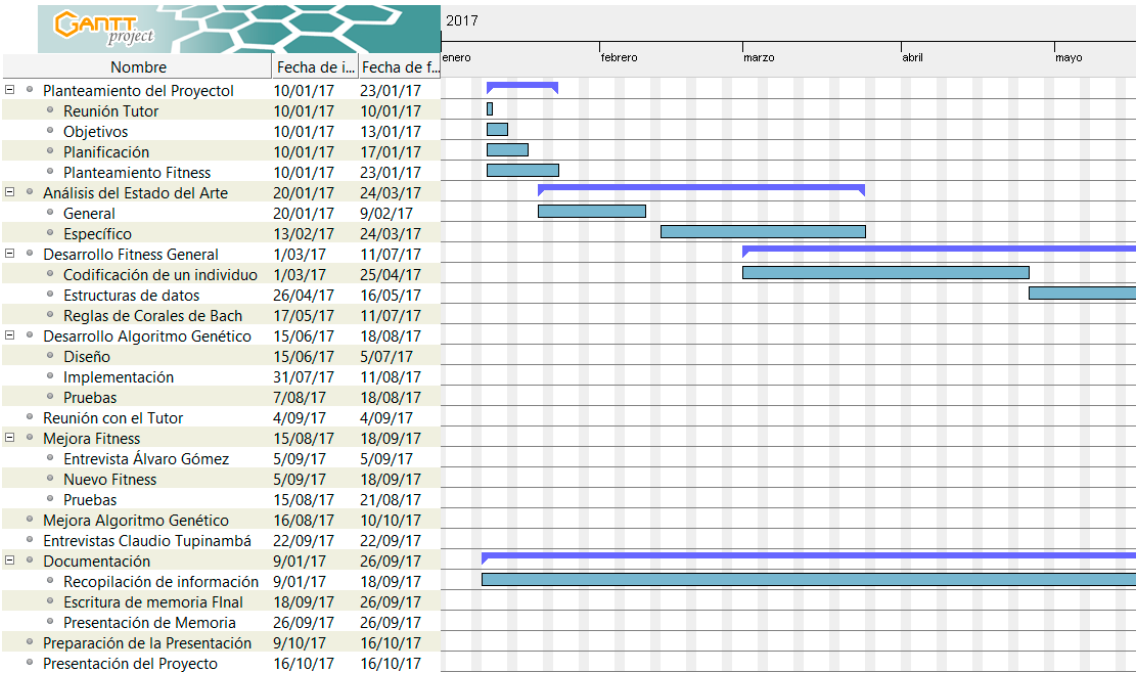
- **Entrevista Final:** Se realiza una entrevista con el maestro compositor Claudio Tupinambá, para preguntarle su opinión acerca de los fines de la investigación y la utilidad en el campo de la música.

- **Documentación:** Se reúne toda la información del proyecto y se documenta en esta memoria.

- **Presentación del proyecto.**

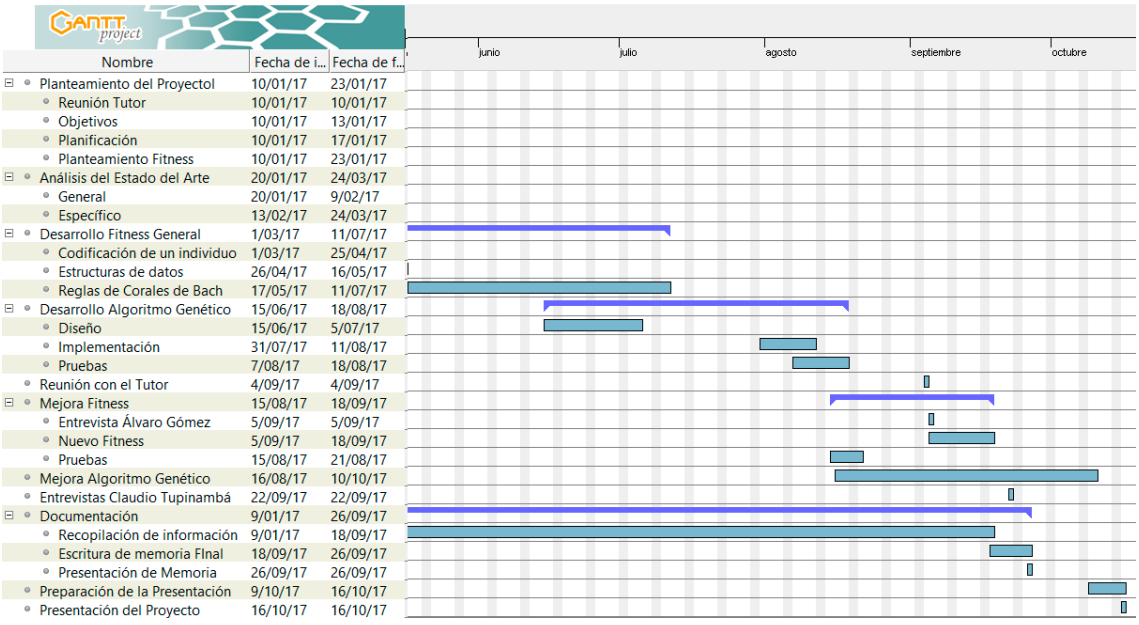
Estas fases se planificaron desde primeros del año 2017 con el objetivo de presentar en Junio de 2017, pero el proyecto sufrió ciertos retrasos debido a la estancia Erasmus del autor y la entrada en el verano, lo que retrasó las comunicaciones y disminuyó las horas disponibles por día para dedicarle a este proyecto.

A sabiendas de todo esto se presenta el diagrama de Gantt que explica la planificación final seguida, realizado mediante el programa “Gantt Project” de licencia pública.



Nombre	Fecha de i...	Fecha de f...
Planteamiento del Proyectol	10/01/17	23/01/17
Reunión Tutor	10/01/17	10/01/17
Objetivos	10/01/17	13/01/17
Planificación	10/01/17	17/01/17
Planteamiento Fitness	10/01/17	23/01/17
Análisis del Estado del Arte	20/01/17	24/03/17
General	20/01/17	9/02/17
Específico	13/02/17	24/03/17
Desarrollo Fitness General	1/03/17	11/07/17
Codificación de un individuo	1/03/17	25/04/17
Estructuras de datos	26/04/17	16/05/17
Reglas de Corales de Bach	17/05/17	11/07/17
Desarrollo Algoritmo Genético	15/06/17	18/08/17
Diseño	15/06/17	5/07/17
Implementación	31/07/17	11/08/17
Pruebas	7/08/17	18/08/17
Reunión con el Tutor	4/09/17	4/09/17
Mejora Fitness	15/08/17	18/09/17
Entrevista Álvaro Gómez	5/09/17	5/09/17
Nuevo Fitness	5/09/17	18/09/17
Pruebas	15/08/17	21/08/17
Mejora Algoritmo Genético	16/08/17	10/10/17
Entrevistas Claudio Tupinambá	22/09/17	22/09/17
Documentación	9/01/17	26/09/17
Recopilación de información	9/01/17	18/09/17
Escritura de memoria FInal	18/09/17	26/09/17
Presentación de Memoria	26/09/17	26/09/17
Preparación de la Presentación	9/10/17	16/10/17
Presentación del Proyecto	16/10/17	16/10/17

Figura 20 - Diagrama de Gantt Parte 1



Nombre	Fecha de i...	Fecha de f...
Planteamiento del Proyectol	10/01/17	23/01/17
Reunión Tutor	10/01/17	10/01/17
Objetivos	10/01/17	13/01/17
Planificación	10/01/17	17/01/17
Planteamiento Fitness	10/01/17	23/01/17
Análisis del Estado del Arte	20/01/17	24/03/17
General	20/01/17	9/02/17
Específico	13/02/17	24/03/17
Desarrollo Fitness General	1/03/17	11/07/17
Codificación de un individuo	1/03/17	25/04/17
Estructuras de datos	26/04/17	16/05/17
Reglas de Corales de Bach	17/05/17	11/07/17
Desarrollo Algoritmo Genético	15/06/17	18/08/17
Diseño	15/06/17	5/07/17
Implementación	31/07/17	11/08/17
Pruebas	7/08/17	18/08/17
Reunión con el Tutor	4/09/17	4/09/17
Mejora Fitness	15/08/17	18/09/17
Entrevista Álvaro Gómez	5/09/17	5/09/17
Nuevo Fitness	5/09/17	18/09/17
Pruebas	15/08/17	21/08/17
Mejora Algoritmo Genético	16/08/17	10/10/17
Entrevistas Claudio Tupinambá	22/09/17	22/09/17
Documentación	9/01/17	26/09/17
Recopilación de información	9/01/17	18/09/17
Escritura de memoria FInal	18/09/17	26/09/17
Presentación de Memoria	26/09/17	26/09/17
Preparación de la Presentación	9/10/17	16/10/17
Presentación del Proyecto	16/10/17	16/10/17

Figura 21 - Diagrama de Gantt Parte 2.

6.4. | Presupuesto.

Para la realización de este proyecto se van a generar gastos procedentes de los recursos materiales y humanos. Los primeros se dividen en equipos y software utilizado, viajes y dietas y otros gastos indirectos. Los recursos humanos hacen referencia a las horas realizadas para cada fase del proyecto y su respectivo coste.

6.4.1. | Software.

Dentro de los recursos de software se incluyen todos los programas y licencias que se han necesitado para realizar este proyecto.

Recurso	Cantidad	Coste (€)	Coste total (€)
Microsoft Office 2013	1	69,00	69,00
GitHub Premium	1 x 6 meses	7,00	42,00
Photoshop CC	1 x 6 meses	19,66	117,96
Sibelius	1 x 6 meses	21,72	130,32
Eclipse Luna	1	0,00	0,00
Python	1	0,00	0,00
Gantt Project	1	0,00	0,00
		TOTAL	359,28

Tabla 31 - Presupuesto: Software.

6.4.2. | Hardware.

Dentro de los recursos de hardware se incluyen todos los equipos utilizados. Sobre el coste total de cada objeto se calcula la amortización del mismo, que consiste en la siguiente fórmula:

$$\frac{N}{T} * C * P$$

Figura 22 - Amortización.

N: meses de uso del equipo. | **T:** Meses de depreciación (12 meses) | **C:** Coste del equipo | **D:** Uso (%)

Recurso	Cantidad	Coste (€)	Meses de Uso	Tiempo de depreciación (meses)	% de uso	Coste amortizado (€)
Ordenador Portatil MSI GE62 6QD Apache Pro	1	1.300,00	6	12	0,95	617,50
Impresora HP	1	52,14	2	12	0,10	0,87
Pendrive 16 GB	1	9,00	4	6	0,20	1,20
					TOTAL	619,57

Tabla 32 - Presupuesto: Hardware.

6.4.3. | Viajes y Dietas.

En este presupuesto se incluyen los viajes para las entrevistas y reuniones realizadas a lo largo del desarrollo del proyecto.

Las entrevistas se realizaron en el Conservatorio Profesional de Música de Leganés, Madrid y las reuniones en la Escuela Politécnica de Leganés de la Universidad Carlos III de Madrid.

El coste es el que supone el viaje en metro entre el domicilio del autor y estas dos localidades.

Reunión	Cantidad	Coste (€)	Coste total (€)
Tutor Yago Sáez	3	3,00	9,00
Entrevista Claudio Tupinambá	2	3,00	6,00
Entrevista Álvaro Israel Gómez Alvarado	1	3,00	3,00
		TOTAL	18,00

Tabla 33 - Presupuesto: Viajes y dietas.

6.4.4. | Gastos Indirectos.

En esta sección se agrupan todos aquellos gastos derivados del proceso de desarrollo, como el material de oficina (bolígrafos, tóner, folios, etc), consumo energético o el gasto de internet.

Concepto	Coste (€/mes)	Periodo de uso (nº meses)	Coste total (€)
Material de Oficina	10,00	6,00	60,00
Internet Movistar	14,90	6,00	89,40
Electricidad	44,90	6,00	269,40
		TOTAL	418,80

Tabla 34 - Gastos Indirectos.

6.4.5. | Recursos Humanos.

El gasto en recursos humanos viene dado una vez se conoce la planificación de proyecto, de ella se puede extraer la siguiente información:

Fase	Días	Media Horas/Día	Horas	Tarea de …
Planteamiento del Proyecto	13	2,00	26	Analista
Análisis del Estado del Arte	63	2,00	126	Analista
Desarrollo Fitness General	132	0,76	100	Analista y programador
Desarrollo Algoritmo Genético	64	0,94	60	Analista y programador
Reunión con el Tutor	1	2,00	2	Analista
Mejora Fitness	34	2,00	68	Analista y programador
Mejora Algoritmo Genético	55	2,00	110	Analista y programador
Entrevista Final	2	2,00	4	Analista
Documentación	-	-	100	Personal de documentación
Presentación (preparación)	8	2,00	16	Autor
		TOTAL	612	

Tabla 35 – RRHH: Horas de trabajo.

Los campos de documentación de días no se han rellenado debido a que la documentación se ha realizado en paralelo durante el desarrollo del proyecto

Una vez se conocen las horas utilizadas para cada fase del proyecto, se multiplican por el coste de cada puesto de trabajo:

- **Analista:** Encargado de las tareas de investigación, diseño de soluciones y análisis de los resultados.
- **Programador:** Implementa la solución y extrae los resultados de las pruebas pedidas por el analista.
- **Persona encargada de la documentación:** Se encarga de todas las tareas de recopilación de información y escritura de los documentos.

Trabajo	€/hora	horas	Total (€)
Analista	12	346	4152,00
Programador	10	150	1500,00
Personal de documentación	10	100	1000,00
Presentación	8	16	128,00
		TOTAL	6780,00

Tabla 36 - Presupuesto: RRHH.

6.4.6 | Presupuesto final.

Una vez se han desglosado todos los gastos, se procede a agruparlos para obtener el presupuesto final del proyecto más el 21% de IVA. De este modo se obtiene que se prevé un gasto total de 8.806,90€.

Concepto	Coste (€)
Software	359,28
Hardware	619,57
Viajes y Dietas	18,00
Gastos Indirectos	418,80
Recursos Humanos	6.780,00
IVA (21%)	1.721,09
TOTAL	9.916,74

Tabla 37 - Presupuesto: Final.

07. | CONCLUSIONES.

Como conclusiones finales del proyecto de investigación que compete a estos documentos, se ha podido comprobar a lo largo del proyecto la profunda carga de tiempo y esfuerzo que conlleva estudiar, analizar y sistematizar todos los parámetros que tienen en cuenta un compositor de Corales de Bach.

Algo que para un compositor experimentado es relativamente sencillo, al afrontarlo bajo los conocimientos de la música evolutiva, con las técnicas y medios computacionales disponibles, se vuelve un trabajo realmente costoso para la máquina.

Esto demuestra, ante todo, la gran necesidad de buscar nuevas líneas de investigación como esta para encontrar las claves lógicas del funcionamiento cognitivo de un cerebro humano a la hora de componer.

Pese no haber conseguido el objetivo de llegar a crear soluciones, satisfactorias tanto musicalmente como computacionalmente hablando, se ha abierto esta nueva senda de investigación.

Se cree que bajo las futuras mejoras de nichos e islas, estás permitirán aumentar la eficiencia lo suficiente para llegar a cumplir los objetivos planteados.

Por último, resaltar que se ha conseguido el objetivo principal del proyecto, que era plasmar la perspectiva de un músico experimentado en la composición, sobre una función de fitness que permita a un algoritmo encontrar soluciones viables.

En la presentación de este proyecto, se espera poder presentar nuevos resultados y mejoras que apoyen todos los avances conseguidos en este proyecto.

08. | FUTURAS LÍNEAS DE INVESTIGACIÓN.

Este proyecto, como ya se ha comentado, busca abrir una nueva línea de investigación, afrontando la música evolutiva siendo lo más fiel posible a la perspectiva de los músicos, pero intentándola sintetizar, en la medida de lo posible.

Las futuras líneas de investigación se pueden dividir en cuatro bloques:

- En lo referente a continuar con el proyecto, éste se puede extender a nuevas formas musicales, como la Fuga o la Sonata. También se podría extender la composición automática de las Corales de Bach introduciendo los conceptos de cadencias, modulaciones, ritmo, notas de paso, etc.

- Por otro lado, se pueden generar herramientas a partir de este proyecto, por ejemplo: a partir del fitness se puede crear una herramienta que corrija los ejercicios de armonía de los alumnos, indicándoles donde ha tenido los fallos y cuáles han sido. También se podría crear una herramienta que haga uso del algoritmo, para que mediante una interfaz gráfica, pueda presentar múltiples soluciones a un ejercicio dado.

- En lo referente al algoritmo genético, se podría implementar y probar el enfoque del diseño desde el concepto de islas, explicado en las mejoras del algoritmo. También se podría buscar más velocidad reduciendo el campo de búsqueda, o incluso se podría estudiar el uso de otras técnicas basadas en la biología, como por ejemplo, los enjambres de hormigas.

- El último bloque consiste en la posible utilización de este algoritmo para añadirle el concepto de interactividad. Esta posible mejora podría consistir en que el propio algoritmo se retro alimentase, por otro lado, que se creara una interfaz que permitiera la colaboración entre el algoritmo con un usuario, de esta formas se conseguiría un apoyo mayor del fitness y unas composiciones adaptadas al gusto del propio usuario.

09. | BIBLIOGRAFÍA.

[1] D. Byrne, Cómo Funciona la Música, Literatura Random House, 2014.

[2] M. D. Borrajo, J. González Boticario y P. Isasi Viñuela, Aprendizaje Automático, Madrid: Sanz y Torres, S.L., 2006.

[3] E. Reck Miranda y J. Al Biles, Evolutionary Computer Music, London: Springer, 2007.

[4] J. Zamacois, Tratado de Armonía., España: Span Press, 1997.

[5] J. P. Burkholder, D. J. Grout y C. V. Palisca, Historia de la música occidental, España: Alianza Música, 2010.

[6] D. Cope, «Experiments In Music Intelligence (EMI),» University of California, Santa Cruz, 1987.

[7] J. A. Biles, «GenJam,» [En línea]. Available: http://igm.rit.edu/~jabics/GenJam.html. [Último acceso: 26 09 2017].

[8] C. Lertora Ginés, P. López de Arenosa Berbeito y J. Zevallos Rodríguez, «MyTone: Compositor musical con algoritmos genéticos.,» Universidad Complutense, Madrid, 2013.

[9] P. Gibson y J. Byrne, «Neurogen,» de *Second International Conference on Artificial Neural Networks.*, Bournemouth, UK, 1991.

[10] Intermorphic, «Intermorphic,» [En línea]. Available: https://intermorphic.com/.[Último acceso: 26 09 2017].

[11] J. A. Biles, «GenJam: Charla xTED,» [En línea]. Available: https://www.youtube.com/watch?v=rFBhwQUZGxg&list=PL89268920295951CC&index=5&feature=plpp_video . [Último acceso: 26 09 2017].

[12] F. Del Carmen Herrera Juárez, K. Peña y P. Malaquias Quintero Flores , «Composición de Sonatas para Piano Mediante,» Instituto Tecnolñogico de Tlaxcala, México..

[13] J. Romero, J. McDermott y J. Correia, «Evolutionary and Biologically Inspired Music, Sound, Art and Design,» de *EvoMUSART*, Granada, España, 2014.

[14] «Music Information Retrieval, Computational Aesthetics, and Artificial Creativity,» [En línea]. Available: http://sger.cs.cofc.edu/.

[15] M. Fukumoto, «Creation of Music Chord Progression Suited for User's Feelings Based on,» Fukuoka, Japan, 2014.

[16] C. B. Sánchez, «GitHub Personal,» [En línea]. Available: https://github.com/CarnsBones. [Último acceso: 26 09 2017].

[17] «Python Foundation,» [En línea]. Available: https://www.python.org/ . [Último acceso: 26 09 2017].

[18] E. Foundation, «Eclipse,» [En línea]. Available: https://eclipse.org/.[Último acceso: 26 09 2017].

[19] PyDev, «PyDev,» [En línea]. Available: http://www.pydev.org/.[Último acceso: 26 09 2017].

[20] Microsoft, «Microsoft Office,» [En línea]. Available: https://products.office.com/es-es/excel . [Último acceso: 26 09 2017].

[21] LillyPond, «LillyPond,» [En línea]. Available: http://lilypond.org . [Último acceso: 26 09 2017].

[22] Abjad, «Abjad,» [En línea]. Available: http://www.projectabjad.org/.[Último acceso: 26 09 2017].

[23] CreativeCommons, «CreativeCommons,» [En línea]. Available: https://creativecommons.org. [Último acceso: 26 09 2017].

10. | ANEXOS.

10.1. | Índice de figuras.

Figura 1 - Esquema general del AG. 13
Figura 2 - Claves musicales 16
Figura 3 - Acordes 17
Figura 4 - Coral de Bach 18
Figura 5 - Neurogen esquema 21
Figura 6 - GenJam 23
Figura 7 - Codificación GenJam 25
Figura 8 - Sonata en árbol. 27
Figura 9 - Input/output 33
Figura 10 - Diseño general de la solución. 46
Figura 11 - Codificación 48
Figura 12 - Selección 49
Figura 13 - Cruces 50
Figura 14 - Proceso de cruce. 50
Figura 15 - Estructuras de datos 51
Figura 16 - Composición fitness V1 68
Figura 17- Composición Fitness V2 71
Figura 18- Algoritmo Genético Con Islas. 85
Figura 19- COmpasición Prueba Final. 87
Figura 20 - Diagrama de Gantt Parte 1 95
Figura 21 - Diagrama de Gantt Parte 2. 95
Figura 22 - Amortización. 96
Figura 23 - Genetic Algorithm 110
Figura 24 - Music Keys. 112
Figura 25 - Chords. 113

Figura 26 - Coral of Bach. 114
Figura 27 - Coding Solution. 117
Figura 28 - Tournament Process..... 118
Figura 29 - Clash Process..... 119

10.2. | Índice de tablas.

tabla 1 - Información de la composición. 36
tabla 2 - Reglas Corales de Bach Parte 1 40
tabla 3 - Progresiones armónicas..... 40
tabla 4 - Relación entre información y estructuras de datos 52
tabla 5 - Tonality: Atributos 53
tabla 6 - Tonality: Métodos..... 53
tabla 7 - Note: Atributos. 54
tabla 8 - Note: Métodos..... 55
tabla 9 - VoicesTessitura: Métodos. 56
tabla 10 - MusicEncoder: Métodos. 56
tabla 11 - Individual: Atributos. 60
Tabla 12 - Invididual: Métodos Parte 1. 62
Tabla 13 - Population: Atributos. 63
Tabla 14 - Population: Métodos. 64
Tabla 15 - Individual: Métodos Parte 2 (fitness V1) 67
Tabla 16 - Fitness: Atributos V1 67
Tabla 17 - Fitness: Métodos..... 67
Tabla 18 - Individual Métodos Parte 3 (fitness V2) 69
Tabla 19 - Fitness: Atributos V2 70
Tabla 20 - Progresión de acordes V2..... 72
Tabla 21 - Individual: Métodos Parte 3 (fitness V3) 73
Tabla 22 - Fitnes Métodos V3 73

Tabla 23 - Fitness Final: Reglas. 79
Tabla 24 - Individual: Métodos Parte 4 (fitness final) 79
Tabla 25 - Fitness Final: Atributos. 80
Tabla 26- Pruebas Tamaño de Población 82
Tabla 27 - Pruebas redundancia codificación. 83
Tabla 28 - Pruebas Tipo Torneo. 83
Tabla 29 - Pruebas Tipo Cruce. 83
Tabla 30 - Pruebas Mutación. 84
Tabla 31 - Presupuesto: Software. 96
Tabla 32 - Presupuesto: Hardware. 97
Tabla 33 - Presupuesto: Viajes y dietas. 97
Tabla 34 - Gastos Indirectos. 98
Tabla 35 – RRHH: Horas de trabajo. 98
Tabla 36 - Presupuesto: RRHH. 99
Tabla 37 - Presupuesto: Final. 99

10.3. | Índice de Gráficas.

Gráfica 1 - Fitness V1 Solo 68
Gráfica 2 - Fitness V2 Sólo 70
Gráfica 3 - Fitness V2 Comparativa 71
Gráfica 4 - Fitness V3 Sólo 73
Gráfica 5 - Fitness V3 Comparativa. 74
Gráfica 6 - Fitness Final Sólo. 81
Gráfica 7 - Fitness Final Comparativa. 81
Gráfica 8 - Prueba Final Comparativa. 86
Gráfica 9 - Prueba Final Sólo. 86
Gráfica 10 - Final Resullt Comparative 120
Gráfica 11 - Final Result. 120

Gráfica 12 - Final Composition..... 121

11. | ENGLISH'S SYNOPSIS.

11.1. | Introduction

In this document is the information regarding the work of end of degree realized by Carlos Bragado Sánchez under the tutoría of the professor / Professor Yago.

The following pages collect all the information of this project: from the origin of the idea and the study of the state of the art; from the analysis of solutions, and developing the most viable to the possible improvements and conclusions extracted throughout the whole process.

11.1.1. | Document's Structure.

This document is structured as follows:

- **Gratefulness.**
- **Biography:** It explains a brief biography of the author.
- **Introduction:** The proposed problem is defined, then explains what are genetic algorithms, music and Bach corals. This allows the reader to be able to defend himself in the terminology dealt with in the document.
- **State of the art:** It explains the process of analysis of the projects and investigations within the field of study that is being treated. It is divided into two parts: a general analysis and a more specific one. With this, it seeks to contextualize the project and relate it to the projects analyzed.
- **Analysis of the problem:** It explains, in detail, the problem raised, possible solutions are presented and argued; in turn, are discussed and explain in detail the objectives and hypotheses to be investigated.
- **Design of the solution:** The selected solution is designed between those analyzed in the analysis, the technical environment is explained that is going to be used. The tests performed for the decisions made in the design are presented and the different improvements are presented from the first design along with the tests that refute them.
- **Final tests:** The tests carried out on the final design of the solution are exposed and critically commented.
- **Project management:** It details the necessary information within the socio-economic environment, the regulatory framework, the planning carried out in the project and the budget of the project.

11.1.2. | Problem Statement.

In ancient Greece it is observed that the philosophers and great sages of the time defended knowledge from the union of fools the types of knowledge.

It was defended that the people that deficaban their life to the knowledge, had to know and have enough skill in all type of subjects.

Throughout history it can be observed that the great sages were masters in many branches of knowledge.

A great example is Leonardo da Vinci, he made contributions to the fields of aerodynamics, anatomy, botany, painting, sculpture, architecture and many others. Some of his best known works are "La Gioconda" or gliders.

It has created a rift between different knowledge, especially between science and art that is considered that there is no union between them.

If you observe the actuality, it is verified as we are returning to the idea of the antiquity. For example, agile methodologies seek to group the different views of varied fields of study.

The union between the different fields of study is completely necessary in the investigation of the artificial intelligence. Artificial intelligence can be defined as the search for the key to reproduce the concept of intelligence with computer techniques.

The path to this objective necessarily involves understanding the functioning of learning different skills, understanding the process of solving different problems, and so on.

There is a philosophical debate about the definition of the concept of intelligence. One of the attributes most attributed to intelligence is the ability to make subjective decisions, that is, an intelligent being is able to feel and make decisions both for and against what he feels to solve problems.

An example of this characteristic is given in the musical creation that is the subject that is spoken about in this project.

Music is understood as a succession of sounds and silences that follow an order created by the composer. The purpose of music is to express an emotion to the listener.

There is a belief that musical creation is necessarily subjective, but this is not entirely true.

David Byrne explains that composers work backwards, as they are not aware that they seek to fit their composition into the backend they have. [1]

This is also demonstrated in the way in which harmony, analysis and composition are imparted in music conservatories. There are very structured and regulated musical forms used, mainly, for the formation of new musicians. The most important examples come from the Baroque with the School Escape, whose parts and form are fixed and strongly regulated. This type of composition is one of those used both in the subjects of analysis and composition to initiate the students to handle complex works.

On the other hand it is necessary that the future professional musician learn Harmony. The key musical form for this learning is Bach's corals, where four voices flow along the staff, complying with very strict norms that allow the learner to learn to understand and analyze the different harmonies and to control both harmonic and melodic movements of the compositions.

Bach's corals are the center of the problem that is going to arise and try to solve by means of interactive genetic algorithms in the next pages.

Therefore, it will begin with an immersion in the context of Bach's genetic algorithms, music and corals.

11.1.3. | Genetic Algorithms.

To begin to talk about genetic algorithms it is necessary to know its history and this leads to the name of John Holland, who establishes the necessary foundations for the development of this method in the 70s; although, it must be remarked, that it does not extend its use in real problems until the 80's.

Genetic algorithms arise from the idea of imitating the various ways of solving problems that exist in nature; more specifically, they are based on Darwin's theorem of evolution, which in a very brief way, explains the evolution of species due to the need for adaptation and survival of the stronger and / or better adapted to the environment in which they inhabit.

Similarly, these algorithms generate an initial population of individuals, solutions to the problem to be treated, and make a selection of the best to that problem to generate offspring that compose a new population of individuals who will go through the selection process again and reproduction.

In this way, the algorithm is getting more and more solutions adapted to the problem.

The use of genetic algorithms is given in the resolution of problems too extensive and complex to be solved by other types of methods since to obtain a solution by these would entail a computational cost and of not feasible time.

General operating diagram:

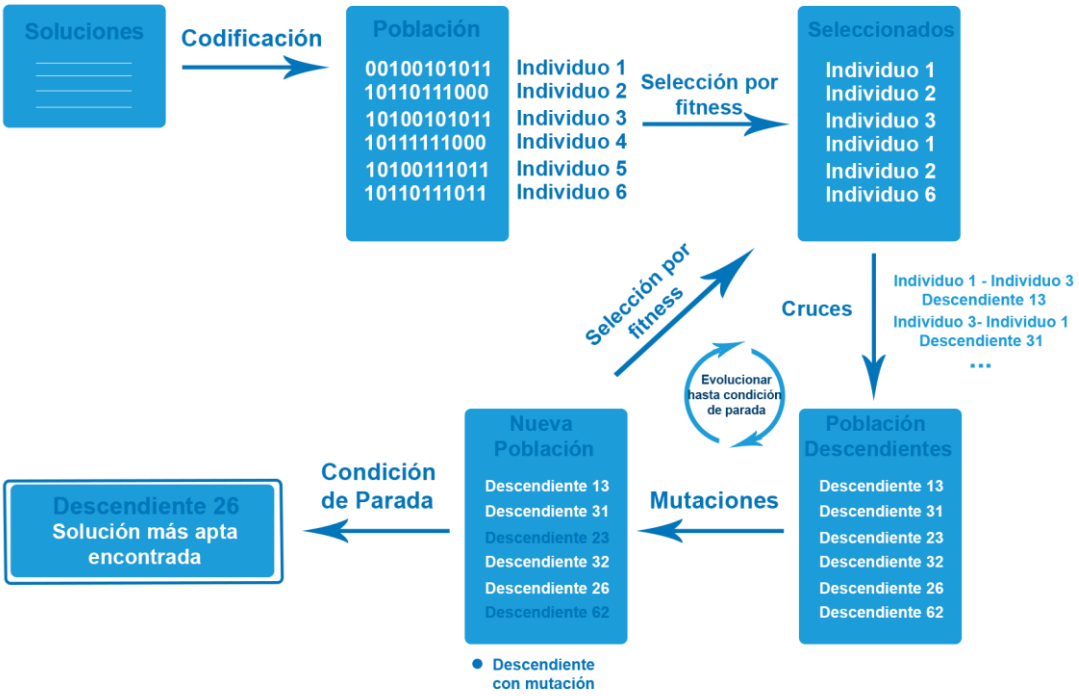


Figura 23 - Genetic Algorithm

This general scheme of operation consists of:

- First, the solutions are encoded by forming a chromosome or individual with each.
- These individuals make up a population.
- It analyzes the degree of adaptation that each individual /solution has to the problem (fitness).
- A selection of the most suitable.
- On this selection are made crosses between them with which a new population is obtained.
- Certain mutations occur in the individuals of the new population, under a limited probability that they occur.
- The process is repeated from the analysis of the fitness of each individual of the new population until the stop condition is fulfilled.
- Finally, the most suitable solution found in the evolution process is returned.

Definitions:

- **Chromosome or individual:** It is a possible solution to the codified problem.

- **Gene**: It consists of each of the parts in which a chromosome is divided. In the case of bit coding, a gene is each of the bits of which the chromosome is composed.

- **Population**: Set of chromosomes or individuals (possible solutions)

- **Initial population**: First population generated in the evolution process.

- **Fitness**: Value that defines the degree of adaptability of an individual to the environment, ie, ability of a solution to solve the problem.

- **Selection**: Process by which individuals with better fitness of the population are chosen. There are different types of selection: by tournament, by roulette, etc. It is necessary to analyze which is more effective in each problem.

- **Crossing or reproduction:** The process in which a new "son / offspring" individual is formed from the gene pool of two or more "parent / ancestor" individuals of the population. There are numerous forms of reproduction that need to be analyzed for each problem, since, depending on the problem, some may work better than others.

- **Mutation:** The process in which a gene changes its value. This may happen in each gene on a chromosome under some probability, usually a very low probability.

- **Stop condition:** Requirement for the evolution process to end. This condition can be from stopping when it is analyzed that the algorithm is stagnant in the improvement, to stop when certain time has passed or certain generations.

- **Generation:** A cycle in the process of evolution, ie, if you observe the scheme shown above, each time you reach "New Population" can be named as the start of a new generation. [2]

11.1.4. | Interactive Genetic Algorithms.

These algorithms work in the same way as those already explained, the big difference is the need for human collaboration in the evolutionary process.

The most frequent interaction is found in fitness, where it is the human who evaluates each individual (for example, in fitness understood as subjective, whether of works of art, literature, music, etc.). Also, interaction can occur in the creation of a database from which to obtain an initial population. [3]

11.1.5. | Music.

Music can be defined as a succession of sounds and silences that follow a harmonious, melodic and rhythmic structure. With the composition of a work is sought to express some kind of feeling to the listener.

As explained, the music consists of:

- **Pentagram**: Set of 5 lines and 4 spaces where the various musical figures are written.

- **Note**: Element that serves to give name to each of the sounds used in the music. They are 7: Do - Re - Mi - Fa - Sol - The – Si

- **Halftone**: Minimum distance between two sounds that are distinguished at different height. It is necessary to add the latter since cultures like Arabic are used quarter notes.

- **Tone**: Distance between two sounds equivalent to two halftones.
- **Alteration**: Symbol to the left of a note (in the writing in pentagram) and indicating that the note will have a sound a semitone above (held "#") or below (aeg "b") of the usual one.
- **Becuadro**: Alteration that represents that the sound of the accompanying note is the usual / natural. It is used when removing one of the other alterations to the note or remembering that it should sound natural.
- **Key**: Figure normally located at the beginning of each staff that indicates the layout of the notes. For example, in the second line Sol key indicates that the Sol note is located on the second line. There are four nomenclatures: Key of Sol, Key of Fa and Key of Do. In turn, the Sol key is always in the second line, the Fa key can be in 4th or 3rd line and the Do key can be in 1st, 3rd or 4th line.

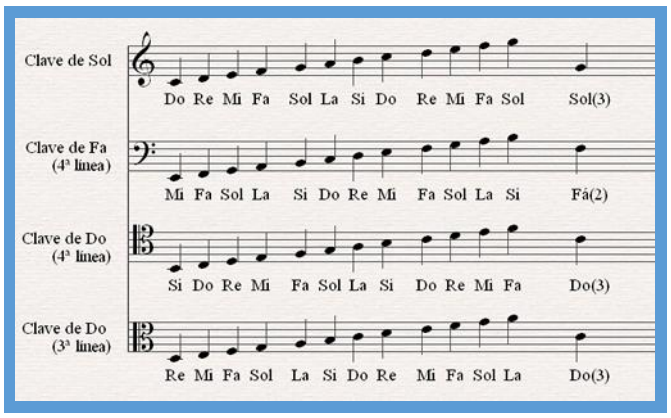


Figura 24 - Music Keys.

- **Octave**: Indicates the level of serious or acute in which a note is found. This project will contemplate four octaves. As you can see in previous figure the last notes are accompanied by a number that indicates in the octave in which they are. This also exemplifies the utility of the keys, by understanding among their notes, which appear within the staff, different octaves between the various keys.
- **Interval**: This is the distance between two notes. It consists of the number of notes between them and, depending on the number of tones and halftones found between them, will be named Major or Righteous, in the case of the 4th, 5th and 8th intervals; these fulfill the distance taken from natural to note in the corresponding interval. Minor, the interval contains a semitone less than the reference from natural. Increased, the range contains one more semitone. Decreased, the range contains a less tone.
- **Harmony**: It is the study of the structure of the notes that are emitted at the same time.
- **Melody**: Succession of notes issued one after another.
- **Chord**: Composition of 3 or more notes issued at the same time.

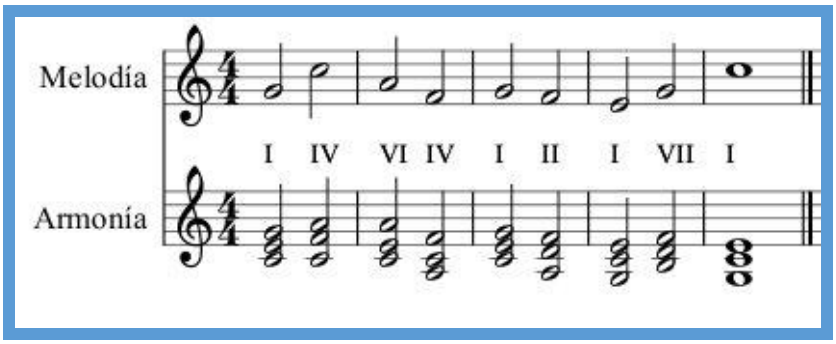


Figura 25 - Chords.

- **Encryption**: Symbols used to designate chord types. In this project will be used the cipher composed of Arabic numerals, which is the standard within classical music. [4]
- **Cadence**: Set of two or three chords that give a certain sense of completion. Example: Perfect cadence between chords V and I.
- **Scale**: Succession of seven notes following the order mentioned above, but if the scale is of Re will be started by said until returning to Re.
- **Tone**: Indicates the alterations that a scale must follow.
- **Armor**: Generally located at the beginning of the staff, consisting of sharp or see, and indicates the tone in which the work is found.

11.1.6. | The Corals of Bach.

In its essence is a Protestant chorus, musical form composed by four voices. The first works are dating back to the 16th century in the Lutheran church and were used in religious ceremonies.

Johann Sebastian Bach (1685 - 1750) is the best known composer of corals and therefore receives his name.

5. Information extracted from the history book of music.

Due to the rigorous rules of composition that must be followed to create a work in this musical form and, in turn, the simplicity of the rhythm in which there is hardly any variation, it has been acquired as a tool of harmony in the professional conservatories of music.

The rules of composition that will be taken into account in this project, make reference to the first course of harmony of the Spanish conservatories.

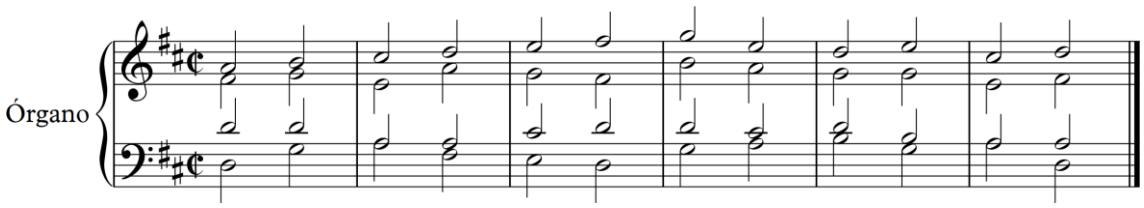


Figura 26 - Coral of Bach.

As can be seen in the picture, Bach's choirs are composed of four voices: the bass is the bass, above the tenor, then the contralto, and the most acute voice is called Soprano or tiple.

11.1.7. | Motivation.

The origin of this project is motivated from the high training, in the professional conservatory, of the author's music.

In an initiative to find the link between studies of computer engineering and music, the idea arises to seek that point of union through genetic algorithms.

The idea of genetic algorithm is to get genes, which alone, can not solve anything excessively complex, be grouped into chromosomes, codify possible solutions to complex problems, and these, in turn, coexist with other chromosomes in populations seeking to create new generations more adapted to the solution.

The author here observes a similarity with the music, the genes can be similar to the notes of a score, by themselves they do not express anything concrete, nevertheless when putting them together in melodies, that would be the chromosomes in the genetic algorithm, they begin to make sense to the work.

These melodies are grouped in harmonies that give a complete expression and coherence to the composition.

Having seen this point of attachment, the desire was that this research could be useful not only in the field of genetic computation, but in the teaching of music.

This is why it is proposed to bring this technique to some of the problems raised in the various subjects of composition.

Seeing it as the first part of a large-scale research project, it has been thought desirable to begin with the subject of harmony and Bach's Corals; which, as already explained, involve a very rigorous composition and, in turn, its composition is simple and strongly structured.

Finally, to emphasize the great utility that can have all the implementation of the musical regulation of the Bach Corals in front of other musical projects and as a great support in the teaching of this musical form, by providing solutions and a possible correction tool.

11.1.7. | Goals.

The main objective of this project is to design a genetic algorithm capable of finding viable solutions to Bach's choral exercises.

More concretely, the algorithm must receive a melody line that refers to the first voice of the choir or soprano, or to the fourth voice or bass. Generally, the first exercises that students tackle are with the voice of the given bass, but at the end of the learning the input corresponds to the soprano voice.

Once the melody line has been received, the algorithm must find solutions that harmonize it into four voices counting the given (bass, tenor, contralto, soprano). These solutions should be evaluated by a fitness that contemplates the rules of the Bach Corals and the positive points that have those that meet all the rules.

Finally, as an end of this process, the results will be analyzed critically looking for conclusions about the algorithm designed.

11.1.8. | Algorithm Design.

The design will be structured in 5 development phases: the general design of the algorithm, the coding of individuals, the selection process, the process of crosses and mutations.

General Design

A random initial population is generated, since it is sought that the algorithm begins from the total ignorance of the problem. The size of this population and the following is initially fixed in 10 individuals; but, the relevant tests will be carried out to see if other sizes give better results.

Once the initial population is available, the process of selecting the most fit for fitness is performed. Of the selected, the relevant crosses are made and a new population is generated with the descendants. These new individuals are mutated under some probability and this gives rise to a new generation that gives way to begin the evolutionary cycle again.

This is done until the stop condition is met, in this case both the manual stop has been used when analyzing the algorithm is stagnant, and the stop in a determined number of generations to compare results with others in the same conditions.

Once the stop condition is fulfilled, the algorithm returns the coded solution in the most apt individual with respect to its fitness found throughout the evolutionary process.

Coding of solutions.

The coding of the solutions must contain the name of the note (do, re, mi, fa, sol, la, si), alteration (#: sustained, b: flat, n: natural) and octave corresponds (1 the octave most serious and 4 the most acute).

It is understood as the coding with fewer redundancies to solve and the more manipulable when it is used to evaluate individuals in the fitness function.

We have not taken into account the sustained doubles, nor the double flats, because they are not used in this type of compositions.

The design of the encoding consists of translating each attribute to bits separately, ie:

- Name of the note: It is coded with three bits (do: 001, re: 010, mi: 011, fa: 100, sol: 101, la: 110, si: 111). The redundancy by empty coding 000 is solved, in this first version, replacing it by "do", that is, "do" is equal to 000 and to 001; this is because it is a very common note. Subsequently tests will be done to use this coding with a random note or with the first note of the tone in which it is being composed, as this is surely the most common note of that composition.

- Alteration name: It is coded with two bits (#: 01, b: 10 and n: 11). The redundancy generated by the empty coding 00, is resolved, a priori as natural, ie 00 and 11 referenced natural. Subsequently, it will be tested, with a random alteration and with the alteration of the first note of the tonality in which it is being composed, since this will be the most common.

- Octave: Finally, the octave is coded with 2 bits being 1 the octave most watered and 4 the most severe one being that, 1: 00, 2: 01, 3:10, 4: 11.

Each note will be encoded with these three attributes so that an example of encoding would be:

Re b 4 = 010(Re) U 10(b) U 11(4) = 0101011 = 7bits

Each chord will be encoded with four notes, that is, by 21 bits. But the coding will be done by voices. That is, first a string of bits is encoded containing all the notes of a voice, then the notes of the next voice and thus with all the voices.

Finally, the individual only takes into account the voice given to evaluate fitness, that is, the coding used for selection, crossing and mutation does not contain the coding of said voice.

Summarized graphically, the coding process is:

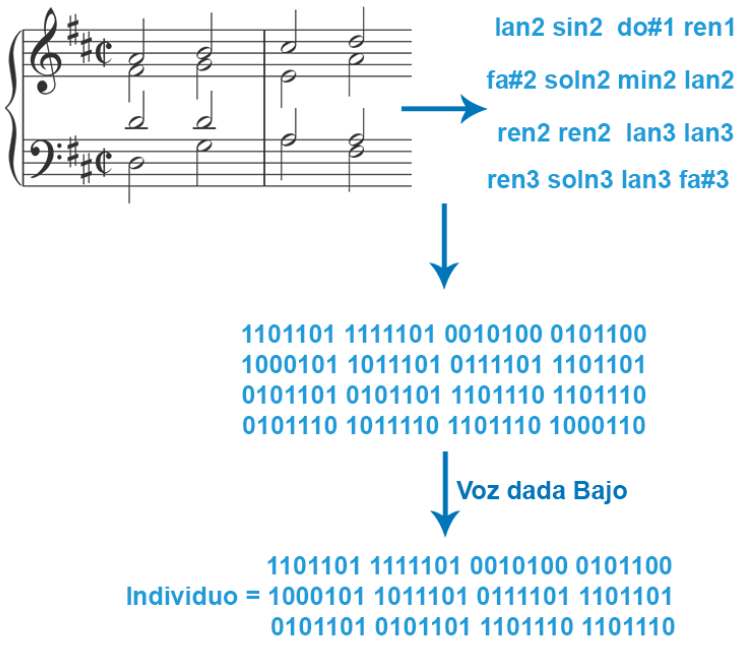


Figura 27 - Coding Solution.

The exercises, based on Bach Chorales, usually contain between 15 and 20 chords, that is, an average of 300 bits per individual. This entails some 2^{300} possible combinations, so it is a codification that necessarily needs a genetic algorithm to find a suitable solution.

Selection operators.

In the design of the selection operations, a selection by tournament has been chosen.

This selection will be made against 3 individuals chosen at random, it is allowed to choose the same individual in different confrontations. Of these 3 individuals will select the best fitness has, that is, the best adapted to the problem.

The decision to define tournaments for 3 individuals is given by the hypothesis that 2 individuals are less likely to eliminate less fit individuals than a tournament with 3.

Figura 28 - Tournament Process.

As shown in the figure in this phase of selection and in the rest of the phases of the evolutionary process, the elitist selection technique is used in which the best individual found so far is replaced by the worst individual in the next population.

Crossing operators

The type of crossover operator that has been designed for this algorithm is the simple crossover; thinking about the idea of thinking of a Bach Chorus as a 3 part structure: introduction, knot and outcome, it has been decided to make this type of crosses on 2 points instead of 1 as usual.

This crossing is performed on two predecessor individuals (X and Y) that are divided into three parts by two randomly selected points. Subsequently, 2 new descendants are created:

- Descendant 1: Part 1 of Y + Part 2 of X + Part 3 of Y.
- Descendant 2: Part 1 of X + Part 2 of Y + Part 3 of X.

Once the new population of descendants is generated, the elite selection is again applied, to replace the worst individual in this new population, for the best found so far.

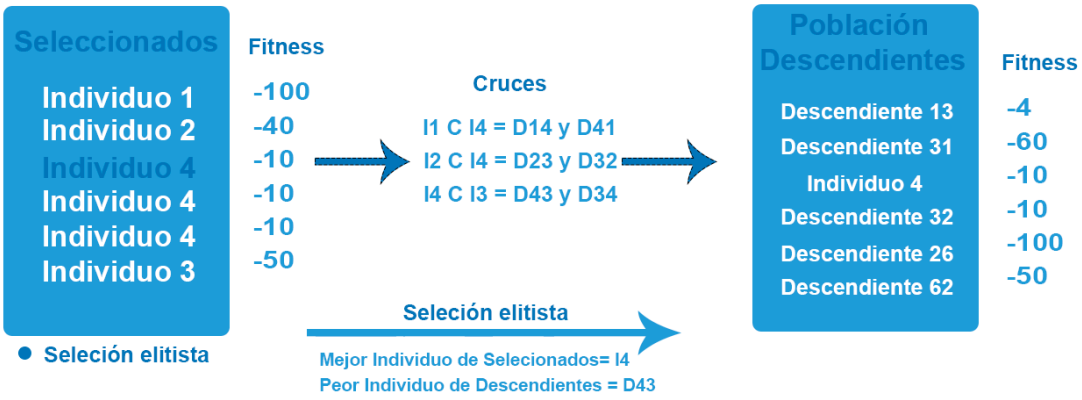


Figura 29 - Clash Process.

Mutation operators.

Como ya se ha comentado en el análisis, la mutación que se va a realizar consiste en la probabilidad de que cambie el valor de cada uno de los bits de un individuo.

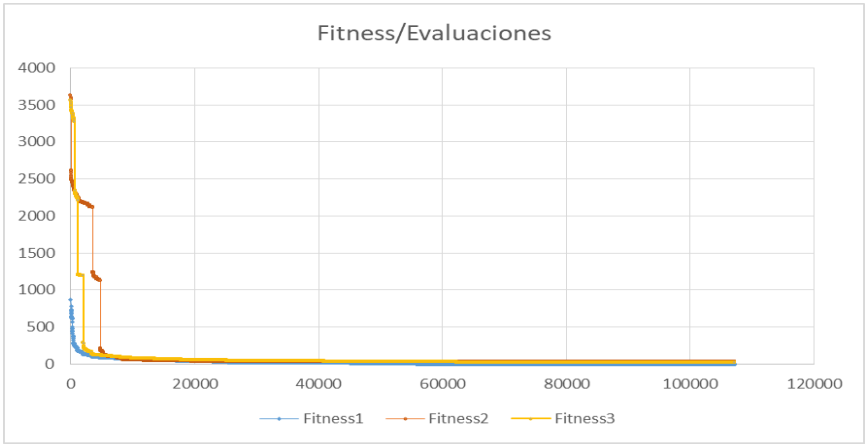
Esta probabilidad se ha establecido en 0.05.

11.1.9. | Results.

As we have analyzed the final design of the algorithm consists of populations of 10 individuals, with selection by tournament of 3, with simple crosses by two points and mutations under a probability of 0.005 of occurrence.

All this provides valid results but, in most tests, inefficient.

We compare all the advances obtained in the following graph:



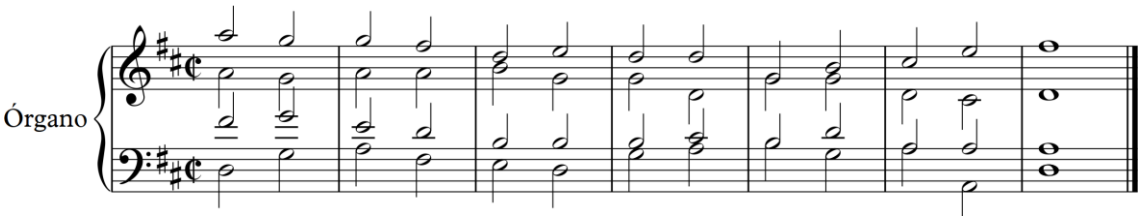
Gráfica 10 - Final Resullt Comparative

The graph of one of the tests on the final design:

Gráfica 11 - Final Result.

An improvement can be seen in the latter tests, but the improvement of efficiency will arise in the project presentation, along with improvements in the concept of niches and / or islands.

Concerning the part of the tests of the musical style:



Gráfica 12 - Final Composition.

An incredible improvement over the first versions of the algorithm can be analyzed. In spite of having a fitness of -12 it captures remarkably better the style of the corals of Bach than those those first results that obtained fitness inferior to -5.

11.1.10. | Conclusions.

As final conclusions of the research project that corresponds to these documents, it has been possible to verify throughout the project the profound time and effort involved in studying, analyzing and systematizing all the parameters that a Bach composer considers.

Something that for an experienced composer is relatively simple, facing it under the knowledge of evolutionary music, with the techniques and available computational means, becomes a really expensive job for the machine.

This shows, above all, the great need to search for new lines of research like this to find the logical keys of the cognitive functioning of a human brain when it comes to composing.

Despite not having achieved the goal of creating solutions, satisfactory both musically and computationally speaking, this new path of research has opened up.

It is believed that under future upgrades of niches and islands, you will allow to increase efficiency enough to reach the objectives set.

Finally, it is important to highlight that the main objective of the project was achieved, which was to capture the perspective of an experienced musician in the composition, on a fitness function that allows an algorithm to find viable solutions.

In the presentation of this project, it is hoped to be able to present new results and improvements that support all the advances achieved in this project.